

**UNIVERSITY OF MUMBAI**  
**No. UG/ 18 of 2016-17**

**CIRCULAR:-**

A reference is invited to the Syllabi relating to the B.Sc. degree course , **vide** this office Circular No. UG/15 of 2016, dated 15<sup>th</sup> June, 2016 and the Principals of affiliated Colleges in Science are hereby informed that the recommendation made by Board of Studies in Science at its meeting held on 23<sup>rd</sup> June, 2016 has been accepted by the Academic Council at its meeting held on, 24<sup>th</sup> June 2016 **vide** item Nos. 4.82 and that the accordance therewith, the revised syllabus of F.Y.B.Sc. (Computer Science) (Sem. I & II), (Credit Based Semester and Grading System ),which is available on the University's web site ([www.mu.ac.in](http://www.mu.ac.in)) and that the same has been brought into force with effect from the academic year 2016-17 which is read as under :-

- 1.The syllabus of F.Y.B.Sc.(Computer Science) is re-structured.
- 2.The Intake capacity of this course as per Govt.of Maharashtra and University of Mumbai's sanction is unchanged.
- 3.The Eligibility for the course is unchanged.
- 4.The fees for the course is Rs.30,000/- per years.

MUMBAI – 400 032  
27<sup>th</sup> June, 2016

*M.A.Khan*  
27/6/16  
(Dr.M.A.Khan)  
REGISTRAR

To,

The Principals of affiliated Colleges in Science.

**A.C/ 4.82/24/06/2016.**

\*\*\*\*\*

No. UG/ 18 of 2016                      MUMBAI-400 032    27<sup>th</sup> June, 2016

Copy forwarded with compliments for information to:-

1. The Dean, Faculty of Science
2. The Director, Board of College and University Development,
3. The Controller of Examinations,
4. The Co-Ordinator, University Computerization Centre.

*M.A.Khan*  
27/6/16.  
(Dr.M.A.Khan)  
REGISTRAR

PTO

**UNIVERSITY OF MUMBAI**



**Syllabus for**  
**Program: Bachelor of Science**  
**Course: Computer Science**

With effect from  
Academic Year 2016-2017

## Preamble

Information and Communication Technology (ICT) has today become integral part of all industry domains as well as fields of academics and research. The industry requirements and technologies have been steadily and rapidly advancing. Organizations are increasingly opting for open source systems. The students too these days are thinking beyond career in the industry and aiming for research opportunities.

The B.Sc. Computer Science course structure therefore needed a fresh outlook and complete overhaul. A real genuine attempt has been made while designing the new syllabus for this 3-year graduate course. Not only does it prepares the students for a career in Software industry, it also motivates them towards further studies and research opportunities.

The core philosophy of overall syllabus is to -

- a. Form strong foundation of Computer science,
- b. Introduce emerging trends to the students in gradual way,
- c. Groom the students for the challenges of ICT industry

In the first year i.e. for semester I & II, basic foundation of important skills required for software development is laid. The syllabus proposes to have four core subjects of Computer science and two core courses of Mathematics-Statistics. All core subjects are proposed to have theory as well as practical tracks. While the Computer Science courses will form fundamental skills for solving computational problems, the Mathematics & Statistics course will inculcate research-oriented acumen.

The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science.

We sincerely believe that any student taking this course will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students' community and teachers' fraternity will appreciate the treatment given to the courses in the syllabus.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents, we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly helped designing certain specialized courses and the syllabus as a whole.

**F.Y.B.Sc. Computer Science Syllabus**  
**Credit Based System and Grading System**  
**Academic year 2016-2017**

Semester – I				
Course Code	Course Type	Course Title	Credits	Lectures/Week
USCS101	Core Subject	Computer Organization and Design	2	3
USCS102	Core Subject	Programming with Python- I	2	3
USCS103	Core Subject	Free and Open Source Software	2	3
USCS104	Core Subject	Database Systems	2	3
USCS105	Core Subject	Discrete Mathematics	2	3
USCS106	Core Subject	Descriptive Statistics and Introduction to Probability	2	3
USCS107	Ability Enhancement Course 1	Soft Skills Development	2	3
USCSP01	Core Subject Practical	Practical of USCS101 + USCS102 + USCS103+USCS104+USCS105+USCS106	6	18

Semester – II				
Course Code	Course Type	Course Title	Credits	Lectures/Week
USCS201	Core Subject	Programming with C	2	3
USCS202	Core Subject	Programming with Python– II	2	3
USCS203	Core Subject	Linux	2	3
USCS204	Core Subject	Data Structures	2	3
USCS205	Core Subject	Calculus	2	3
USCS206	Core Subject	Statistical Methods and Testing of Hypothesis	2	3
USCS207	Ability Enhancement Course 2	Green Technologies	2	3
USCSP02	Core Subject Practical	Practical of USCS201 + USCS202 + USCS203+USCS204+USCS205+USCS206	6	18

## Semester I – Theory

<b>Course:</b>  USCS101	<b>Computer Organization and Design</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objectives.</b> To understand the structure and operation of modern processors and their instruction sets</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) To learn about how computer systems work and underlying principles</li> <li>2) To understand the basics of digital electronics needed for computers</li> <li>3) To understand the basics of instruction set architecture for reduced and complex instruction sets</li> <li>4) To understand the basics of processor structure and operation</li> <li>5) To understand how data is transferred between the processor and I/O devices</li> </ol>		
Unit I	<p><b>Computer Abstractions and Technology:</b> Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters.</p> <p><b>Logic circuits and functions:</b> Combinational circuits and functions: Basic logic gates and functions, truth tables; logic circuits and functions. Minimization with Karnaugh maps. Synthesis of logic functions with and-or-not gates, nand gates, nor gates. Fan-in and fan-out requirements; tristate buffers. Half adder, full adder, ripple carry adder. (Flip flops) Gated S-R and D latches, edge-triggered D latch. Shift registers and registers. Decoders, multiplexers.</p> <p>Sequential circuits and functions: State diagram and state table; finite state machines and their synthesis.</p>	15 L
Unit II	<p><b>Instruction set architectures:</b> Memory organization, addressing and operations; word size, big-endian and little-endian arrangements. Instructions, sequencing. Instruction sets for RISC and CISC (examples Altera NIOS II and Freescale ColdFire). Operand addressing modes; pointers; indexing for arrays. Machine language, assembly language, assembler directives. Function calls, processor runtime stack, stack frame. Types of machine instructions: arithmetic, logic, shift, etc. Instruction sets, RISC and CISC examples.</p>	15 L
Unit III	<p><b>Basic Processor Unit:</b> Main components of a processor: registers and register files, ALU, control unit, instruction fetch unit, interfaces to instruction and data memories. Datapath. Instruction fetch and execute; executing arithmetic/logic, memory access and branch instructions; hardwired and microprogrammed control for RISC and CISC.</p> <p><b>Basic I/O:</b> Accessing I/O devices, data transfers between processor and I/O devices. Interrupts and exceptions: interrupt requests and processing.</p>	15 L

<p>Text book:</p> <ol style="list-style-type: none"> <li>1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012</li> </ol> <p>Additional References:</p> <ol style="list-style-type: none"> <li>1. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011</li> <li>2. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4<sup>th</sup> Edition, 2010</li> </ol>	
--	--

<b>Course:</b> <b>USCS102</b>	<b>Programming with Python- I</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objectives</b> The objective of this paper is to introduce various concepts of programming to the students using Python.</p> <p><b>Expected learning outcomes</b></p> <ol style="list-style-type: none"> <li>1) Students should be able to understand the concepts of programming before actually starting to write programs.</li> <li>2) Students should be able to develop logic for Problem Solving.</li> <li>3) Students should be made familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.</li> <li>4) Students should be able to apply the problem solving skills using syntactically simple language i.e. <b>Python (version: 3.X or higher)</b></li> </ol>		
Unit I	<p>Reasons for Python as the learner’s first programming language. Introduction to the IDLE interpreter (shell) and its documentation. Expression evaluation: similarities and differences compared to a calculator; expressions and operators of types int, float, boolean. Built-in function type. Operator precedence.</p> <p>Enumeration of simple and compound statements. The expression statement. The assert statement, whose operand is a boolean expression (values true or false). The assignment statement, dynamic binding of names to values, (type is associated with data and not with names); automatic and implicit declaration of variable names with the assignment statement; assigning the value None to a name. The del (delete) statement. Input/output with print and input functions. A statement list (semicolon-separated list of simple statements on a single line) as a single interpreter command. The import statement for already-defined functions and constants. The augmented assignment statement. The built-in help() function.</p> <p>Interactive and script modes of IDLE, running a script, restarting the shell.</p> <p>The compound statement def to define functions; the role of indentation for delimiting the body of a compound statement; calling a previously defined function. Compound data types str, tuple and list (enclosed in quotes, parentheses and brackets, respectively). Indexing individual elements within these types. Strings and tuples are immutable, lists are mutable. Built-in functions min, max, sum. Interactive solution of model problems, (e.g., finding the square root of a number or zero of a function), by repeatedly executing the body of a loop (where the body is a statement list).</p>	15 L

Unit II	Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables. The range function, the iterative for statement. The conditional statements if, if-else, if-elif-else. The iterative statements while, while-else, for-else. The continue statement to skip over one iteration of a loop, the break statement to exit the loop. Nested compound statements. Dictionaries: concept of key-value pairs, techniques to create, update and delete dictionary items. Problem-solving using compound types and statements.	15 L
Unit III	Anonymous functions. List comprehensions. Gentle introduction to object-oriented programming; using the built-in dir() function, enumerate the methods of strings, tuples, lists, dictionaries. Using these methods for problem-solving with compound types.	15 L

**Text books:**

1. Magnus Lie Hetland, *Beginning Python: From Novice to Professional*, Apress
2. Paul Gries, et al., *Practical Programming: An Introduction to Computer Science Using Python 3*, Pragmatic Bookshelf, 2/E 2014

**Additional References:**

1. Charles Dierbach, *Introduction to Computer Science using Python*, Wiley, 2013
2. Paul Gries, Jennifer Campbell, Jason Montojo, *Practical Programming: An Introduction to Computer Science Using Python 3*, Pragmatic Bookshelf, 2/E 2014
3. Adesh Pandey, *Programming Languages – Principles and Paradigms*, Narosa, 2008

<b>Course:</b> <b>USCS103</b>	<b>Free and Open-source Software</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objective:</b> Open Source has acquired a prominent place in software industry. Having knowledge of Open Source and its related technologies is an essential for Computer Science student. This course introduces Open Source methodologies and ecosystem to students.</p> <p><b>Expected Learning Outcome:</b></p> <ol style="list-style-type: none"> <li>1) Upon completion of this course, students should have a good working knowledge of Open Source ecosystem, its use, impact and importance.</li> <li>2) This course shall help student to learn Open Source methodologies, case studies with real life examples.</li> </ol>		
Unit I	<p><b>Introduction</b> Introduction: Open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project.</p> <p><b>Methodologies</b> Open Source History, Initiatives, Principle and methodologies. Philosophy : Software Freedom, Open Source Development Model Licenses and Patents: What Is A License, Important FOSS Licenses (Apache,BSD,GPL, LGPL), copyrights and copy lefts, Patents Economics of FOSS : Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization</p>	<b>15L</b>

	<p><b>Social Impact</b>  Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government.</p>	
Unit II	<p><b>Case Studies</b>  Example Projects: Apache web server, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, wordpress, GCC, GDB, github, Open Office. Study: Understanding the developmental models, licensings, mode of funding, commercial/non-commercial use. Open Source Hardware, Open Source Design, Open source Teaching. Open source media.</p> <p><b>Collaboration, Community and Communication</b></p> <p><b>Contributing to Open Source Projects</b>  Introduction to github, interacting with the community on github, Communication and etiquette, testing open source code, reporting issues, contributing code.  Introduction to wikipedia, contributing to Wikipedia Or contributing to any prominent open source project of student's choice.  Starting and Maintaining own Open Source Project.</p>	15L
Unit III	<p><b>Understanding Open Source Ecosystem</b>  Open Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, debuggers, Programming languages, LAMP, Open Source database technologies</p>	15L
<p><b>Text books:</b></p> <ol style="list-style-type: none"> <li>1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006</li> <li>2. The official Ubuntu Book, 8<sup>th</sup> Edition</li> </ol> <p><b>Additional references:</b></p> <ol style="list-style-type: none"> <li>1. The Linux Documentation Project: <a href="http://www.tldp.org/">http://www.tldp.org/</a></li> <li>2. Docker Project Home: <a href="http://www.docker.com">http://www.docker.com</a></li> <li>3. Linux kernel Home: <a href="http://kernel.org">http://kernel.org</a></li> <li>4. Open Source Initiative: <a href="https://opensource.org/">https://opensource.org/</a></li> <li>5. Linux Documentation Project: <a href="http://www.tldp.org/">http://www.tldp.org/</a></li> <li>6. Wikipedia: <a href="https://en.wikipedia.org/">https://en.wikipedia.org/</a></li> <li>7. <a href="https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia">https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia</a></li> <li>8. Github: <a href="https://help.github.com/">https://help.github.com/</a></li> <li>9. The Linux Foundation: <a href="http://www.linuxfoundation.org/">http://www.linuxfoundation.org/</a></li> </ol>		



<b>Course:</b> <b>USCS104</b>	<b>Database Systems</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objectives:</b>  The objective of this course is to introduce the concept of the DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases</p> <p><b>Expected Learning Outcomes</b></p> <ol style="list-style-type: none"> <li>1) Students should be able to evaluate business information problem and find the requirements of a problem in terms of data.</li> <li>2) Students should be able to design the database schema with the use of appropriate data types for storage of data in database.</li> <li>3) Students should be able to create, manipulate, query and back up the databases.</li> </ol>		
Unit I	<p><b>Introduction to DBMS</b> – Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture</p> <p><b>Data models</b> - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model ( relational, hierarchical, network)</p> <p><b>Entity Relationship Model</b> - Entities, attributes, entity sets, relations, relationship sets, Additional constraints ( key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER ( entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER)</p> <p><b>Relational data model</b>– Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint</p> <p><b>ER to Table</b>- Entity to Table, Relationship to tables with and without key constraints.</p>	15L
Unit II	<p><b>Schema refinement and Normal forms:</b> Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition.</p> <p><b>Relational Algebra</b> operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division)</p> <p><b>DDL Statements</b> - Creating Databases, Using Databases, datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables, Backing Up and Restoring databases</p> <p><b>DML Statements</b> – Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause</p>	15L

Unit III	<p><b>Functions</b> – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse)</p> <p><b>Joining Tables</b> – inner join, outer join (left outer, right outer, full outer)</p> <p><b>Subqueries</b> – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries</p> <p><b>Database Protection:</b> Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control</p> <p><b>Views</b> (creating, altering dropping, renaming and manipulating views)</p> <p><b>DCL Statements</b> (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges)</p>	15L
<p><b>Text books:</b></p> <ol style="list-style-type: none"> <li>1. Ramez Elmasri &amp; Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010</li> <li>2. Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill, 2007</li> <li>3. Joel Murach, Murach’s MySQL, Murach, 2012</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press, 2005.</li> </ol>		

<p><b>Course:</b> USCS105</p>	<p><b>Discrete Mathematics</b> (Credits : 2 Lectures/Week: 3)</p>	
<p><b>Objectives:</b> The purpose of the course is to familiarize the prospective learners with mathematical structures that are fundamentally discrete. This course introduces sets and functions, forming and solving recurrence relations and different counting principles. These concepts are useful to study or describe objects or problems in computer algorithms and programming languages.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) To provide overview of theory of discrete objects, starting with relations and partially ordered sets.</li> <li>2) Study about recurrence relations, generating function and operations on them.</li> <li>3) Give an understanding of graphs and trees, which are widely used in software.</li> <li>4) Provide basic knowledge about models of automata theory and the corresponding formal languages.</li> </ol>		

Unit I	<p><b>Recurrence Relations</b></p> <p><b>(a) Functions:</b> Definition of function. Domain, co domain and the range of a function. Direct and inverse images. Injective, surjective and bijective functions. Composite and inverse functions.</p> <p><b>(b) Relations:</b> Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Daigrams , Maximum and Minimum elements, Lattices</p> <p><b>(c) Recurrence Relations:</b> Definition of recurrence relations, Formulating recurrence relations, solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients. Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root, Particular solutions of non linear homogeneous recurrence relation, Solution of recurrence relation by the method of generation functions, Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi, Intersection of lines in a plane, Sorting Algorithms.</p>	15L
Unit II	<p><b>Counting Principles , Languages and Finite State Machine</b></p> <p><b>(a) Permutations and Combinations:</b> Partition and Distribution of objects, Permutation with distinct and indistinct objects, Binomial numbers, Combination with identities: Pascal Identity, Vandermonde’s Identity, Pascal triangle, Binomial theorem, Combination with indistinct objects.</p> <p><b>(b) Counting Principles:</b> Sum and Product Rules, Two-way counting, Tree diagram for solving counting problems, Pigeonhole Principle (without proof); Simple examples, Inclusion Exclusion Principle (Sieve formula) (Without proof).</p> <p><b>(c) Languages, Grammars and Machines:</b> Languages , regular Expression and Regular languages, Finite state Automata, grammars, Finite state machines, Gödel numbers, Turing machines.</p>	15L
Unit III	<p><b>Graphs and Trees</b></p> <p><b>(a) Graphs :</b> Definition and elementary results, Adjacency matrix, path matrix, Representing relations using diagraphs, Warshall’s algorithm- shortest path , Linked representation of a graph, Operations on graph with algorithms - searching in a graph; Insertion in a graph, Deleting from a graph, Traversing a graph- Breadth-First search and Depth-First search.</p> <p><b>(b) Trees:</b> Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees, representing binary trees in memory, traversing binary trees, binary search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree</p>	15L

**Textbook:**

1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)
2. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.
3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc.

**Additional References:**

1. Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition .
2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
3. Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc.
4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi.

<b>Course:</b> <b>USCS106</b>	<b>Descriptive Statistics and Introduction to Probability</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<b>Objectives:</b> The purpose of this course is to familiarize students with basics of Statistics. This will be essential for prospective researchers and professionals to know these basics.		
<b>Expected Learning Outcomes:</b> 1) Enable learners to know descriptive statistical concepts 2) Enable study of probability concept required for Computer learners		
Unit I	<b>Data Presentation</b> Data types : attribute, variable, discrete and continuous variable Data presentation : frequency distribution, histogram o give, curves, stem and leaf display <b>Data Aggregation</b> Measures of Central tendency: Mean, Median, mode for raw data, discrete, grouped frequency distribution. Measures dispersion: Variance, standard deviation, coefficient of variation for raw data, discrete and grouped frequency distribution, quartiles, quantiles Real life examples	15L
Unit II	<b>Moments:</b> raw moments, central moments, relation between raw and central moments <b>Measures of Skewness and Kurtosis:</b> based on moments, quartiles, relation between mean, median, mode for symmetric, asymmetric frequency curve. <b>Correlation and Regression:</b> bivariate data, scatter plot, correlation, nonsense correlation, Karl pearson's coefficients of correlation, independence. <b>Linear regression:</b> fitting of linear regression using least square regression, coefficient of determination, properties of regression coefficients (only statement)	15L

Unit III	<p><b>Probability</b> : Random experiment, sample space, events types and operations of events</p> <p><b>Probability definition</b> : classical, axiomatic, Elementary Theorems of probability (without proof)</p> <ul style="list-style-type: none"> <li>– <math>0 \leq P(A) \leq 1</math>,</li> <li>– <math>P(A \cup B) = P(A) + P(B) - P(A \cap B)</math></li> <li>– <math>P(A') = 1 - P(A)</math></li> <li>– <math>P(A) \leq P(B)</math> if <math>A \subset B</math></li> </ul> <p>Conditional probability, 'Bayes' theorem, independence, Examples on Probability</p>	15L
----------	--	-----

**Text Book:**

1. Trivedi, K.S.(2001) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi

**Additional References:**

1. Ross, S.M. (2006): A First course in probability. 6<sup>th</sup> Ed<sup>n</sup> Pearson
2. Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): common statistical tests. Satyajeet Prakashan, Pune
3. Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi
4. Gupta, S.C. and Kapoor, V.K. (1999): Applied Statistics, S. Chand and Son's, New Delhi
5. Montgomery, D.C. (2001): Planning and Analysis of Experiments, wiley.

<b>Course:</b> USCS107	<b>Soft Skills Development</b> (Credits : 2 Lectures/Week: 3)	
<p><b>Objectives:</b> To help learners develop their soft skills and develop their personality together with their technical skills. Developing professional, social and academic skills to harness hidden strengths, capabilities and knowledge equip them to excel in real work environment and corporate life. Understand various issues in personal and profession communication and learn to overcome them</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) To know about various aspects of soft skills and learn ways to develop personality</li> <li>2) Understand the importance and type of communication in personal and professional environment.</li> <li>3) To provide insight into much needed technical and non-technical qualities in career planning.</li> <li>4) Learn about Leadership, team building, decision making and stress management</li> </ol>		

Unit I	<p><b>Introduction to Soft Skills and Hard Skills</b></p> <p><b>Personality Development:</b> Knowing Yourself, Positive Thinking, Johari's Window, Communication Skills, Non-verbal Communication, Physical Fitness</p> <p><b>Emotional Intelligence:</b> Meaning and Definition, Need for Emotional Intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence, Competencies of Emotional Intelligence, Skills to Develop Emotional Intelligence</p> <p><b>Etiquette and Mannerism:</b> Introduction, Professional Etiquette, Technology Etiquette</p> <p><b>Communication Today:</b> Significance of Communication, GSC's 3M Model of Communication, Vitality of the Communication Process, Virtues of Listening, Fundamentals of Good Listening, Nature of Non-Verbal Communication, Need for Intercultural Communication, Communicating Digital World</p>	15L
Unit II	<p><b>Academic Skills</b></p> <p><b>Employment Communication:</b> Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter</p> <p><b>Professional Presentation:</b> Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation</p> <p><b>Job Interviews:</b> Introduction, Importance of Resume, Definition of Interview, Background Information, Types of Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, Changes in the Interview Process, FAQ During Interviews</p> <p><b>Group Discussion:</b> Introduction, Ambience/Seating Arrangement for Group Discussion, Importance of Group Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits</p>	15L
Unit III	<p><b>Professional Skills</b></p> <p><b>Creativity at Workplace:</b> Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method</p> <p><b>Ethical Values:</b> Ethics and Society, Theories of Ethics, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics</p> <p><b>Capacity Building: Learn, Unlearn and Relearn:</b> Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building</p> <p><b>Leadership and Team Building:</b> Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams,</p> <p><b>Decision Making and Negotiation:</b> Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts</p> <p><b>Stress and Time Management:</b> Stress, Sources of Stress, Ways to Cope with Stress</p>	15L

**Text book:**

1. *Soft Skills: an Integrated Approach to Maximise Personality*, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India

**Additional References:**

1. *Personality Development and Soft Skills*, Barun K. Mitra, Oxford Press
2. *Business Communication*, Shalini Kalia, Shailja Agrawal, Wiley India
3. *Soft Skills - Enhancing Employability*, M. S. Rao, I. K. International
4. *Cornerstone: Developing Soft Skills*, Sherfield, Pearson India

**Semester I – Practical**

Course: USCSP1	Practical of USCS101 + USCS102 + USCS103+USCS104+USCS105+USCS106 (Credits : 6, Lectures/Week: 18)	
USCSP101	<p><b>Computer Organization and Design</b></p> <ol style="list-style-type: none"> <li>1. Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).</li> <li>2. Simplify given Boolean expression and realize it.</li> <li>3. Design and verify a half/full adder</li> <li>4. Design and verify half/full subtractor</li> <li>5. Design a 4 bit magnitude comparator using combinational circuits.</li> <li>6. Design and verify the operation of flip-flops using logic gates.</li> <li>7. Verify the operation of a counter.</li> <li>8. Verify the operation of a 4 bit shift register</li> <li>9. Using SPIM, write and test an adding machine program that repeatedly reads in integers and adds them into a running sum. The program should stop when it gets an input that is 0, printing out the sum at that point.</li> <li>10. Using SPIM, write and test a program that reads in a positive integer using the SPIM system calls. If the integer is not positive, the program should terminate with the message "Invalid Entry"; otherwise the program should print out the names of the digits of the integers, delimited by exactly one space. For example, if the user entered "528," the output would be "Five Two Eight."</li> </ol> <p># Practical No. 1 to 8 can be performed using any open source simulator (like Logisim) (Download it from <a href="https://sourceforge.net/projects/circuit/">https://sourceforge.net/projects/circuit/</a>)</p> <p># Practical No. 9 and 10 are required to be done using SPIM. SPIM is a self-contained simulator that will run MIPS R2000/R3000 assembly language programs.</p> <p># Latest version is available at <a href="https://sourceforge.net/projects/spimsimulator/">https://sourceforge.net/projects/spimsimulator/</a></p>	

<b>USCSP102</b>	<p><b>Programming with Python – I</b></p> <ol style="list-style-type: none"> <li>1. Installing and setting up the Python IDLE interpreter. Executing simple statements like expression statement (numeric and Boolean types), assert, assignment, delete statements; the print function for output.</li> <li>2. Script and interactive modes; defining a function in the two modes; executing a script; interactively executing a statement list (semicolon-separated sequence of simple statements); the input function.</li> <li>3. Programs based on lists, conditional constructs, the for statement and the range function; interactively using the built-in functions len, sum, max, min</li> <li>4. Programs related to string manipulation</li> <li>5. Programs based on the while statement; importing and executing built-in functions from the time, math and random modules</li> <li>6. Programs using break and continue statements.</li> <li>7. Programs related to dictionaries</li> <li>8. Programs using list comprehensions and anonymous functions</li> <li>9. Programs using the built-in methods of the string, list and dictionary classes</li> </ol>	
-----------------	---	--



<b>USCSP103</b>	<p><b>Free and Open Source Software</b></p> <ol style="list-style-type: none"> <li>1. Identify any Open Source software and create detailed report about it. Sample Guidelines.       <ol style="list-style-type: none"> <li>a. Idea</li> <li>b. What problem does it solves?</li> <li>c. Licensing model</li> <li>d. Intent behind making it open source</li> <li>e. Monetization models</li> <li>f. Popularity</li> <li>g. Impact</li> </ol> </li> <li>2. Learn at least three different open source licenses and create a brief report about them.       <ol style="list-style-type: none"> <li>a. History of license</li> <li>b. Idea</li> <li>c. What problems does it solve?</li> <li>d. Detailed licensing model</li> <li>e. Which popular software are released under this license?</li> <li>f. Any popular news associated with this license?</li> <li>g. Popularity</li> <li>h. Impact</li> </ol> </li> <li>3. Contributing to Open Source       <ol style="list-style-type: none"> <li>a. Identify any Open Source project of your interest</li> <li>b. Learn more about the project w.r.t. Lab 1.</li> <li>c. Start contributing to the project either by           <ol style="list-style-type: none"> <li>i. Testing</li> <li>ii. Reporting bugs</li> <li>iii. Coding</li> <li>iv. Helping in documentation</li> <li>v. Participating in discussions</li> <li>vi. Participating in pre-release testing programs</li> <li>vii. UI development.</li> <li>viii. Or any other important area.</li> </ol> </li> </ol> </li> <li>4. Hands on with Open Source Software       <ol style="list-style-type: none"> <li>a. Identify any open source software of your interest</li> <li>b. Learn it from practical view-point</li> <li>c. Give a brief presentation about it to the class</li> <li>d. Sample projects: gcc, gdb, drupal, wordpress, apache web server, mysql database</li> </ol> </li> <li>5. Contributing to Wikipedia:       <ol style="list-style-type: none"> <li>a. Introduction to wikipedia: operating model, license, how to contribute?</li> <li>b. Create your user account on wikipedia</li> <li>c. Identify any topic of your choice and contribute the missing information</li> </ol> </li> <li>6. Github       <ol style="list-style-type: none"> <li>a. Create and publish your own open source project: Write any simple program using your choice of programming language.</li> </ol> </li> </ol>	
-----------------	--	--

<p><b>USCSP103</b></p>	<ul style="list-style-type: none"> <li>b. Create a repository on github and save versions of your project. You'll learn about the staging area, committing your code, branching, and merging,</li> <li>c. Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi-developer projects. You'll learn how to make and review a pull request on GitHub.</li> <li>d. Contribute to a Live Project: Students will publish a repository containing their reflections from the course and submit a pull request.</li> </ul> <p>7. Open Source Operating Systems</p> <ul style="list-style-type: none"> <li>a. Learn any open source operating system of your choice : Linux, Android, FreeBSD, Open Solaris etc.</li> <li>b. Learn the installation.</li> <li>c. Identify the unique features of the OS of your choice.</li> </ul> <p>8. Virtualization: Open Source virtualization technologies:</p> <ul style="list-style-type: none"> <li>a. Install and configure any one: VirtualBox, Zen, KVM</li> <li>b. Create and use virtual machines</li> </ul> <p>9. Containerization:</p> <ul style="list-style-type: none"> <li>a. Containerization technologies: docker, rocket, LXD</li> <li>b. Install and configure any containerization technology</li> <li>c. Create and use containers using it</li> </ul> <p>10. Linux Kernel: Learn Linux kernel with respect to:</p> <ul style="list-style-type: none"> <li>a. What is Linux kernel?</li> <li>b. Operating model</li> <li>c. Licensing Model</li> <li>d. How development works?</li> <li>e. Download kernel source code.</li> <li>f. Compile the Kernel</li> </ul>	
<p><b>USCSP104</b></p>	<p><b>Database Systems</b></p> <ul style="list-style-type: none"> <li>1. For given scenario <ul style="list-style-type: none"> <li>• Draw E-R diagram and convert entities and relationships to table.</li> </ul> </li> <li>2. Write relational algebra queries on the tables created in Practical-1.</li> <li>3. Perform the following: <ul style="list-style-type: none"> <li>• Viewing all databases</li> <li>• Creating a Database</li> <li>• Viewing all Tables in a Database</li> <li>• Creating Tables (With and Without Constraints)</li> <li>• Inserting/Updating/Deleting Records in a Table</li> <li>• Saving (Commit) and Undoing (rollback)</li> </ul> </li> <li>4. Perform the following: <ul style="list-style-type: none"> <li>• Altering a Table</li> <li>• Dropping/Truncating/Renaming Tables</li> <li>• Backing up / Restoring a Database</li> </ul> </li> </ul>	

	<ol style="list-style-type: none"> <li>5. Perform the following: <ul style="list-style-type: none"> <li>• Simple Queries</li> <li>• Simple Queries with Aggregate functions</li> <li>• Queries with Aggregate functions (group by and having clause)</li> </ul> </li> <li>6. Queries involving <ul style="list-style-type: none"> <li>• Date Functions</li> <li>• String Functions</li> <li>• Math Functions</li> </ul> </li> <li>7. Join Queries <ul style="list-style-type: none"> <li>• Inner Join</li> <li>• Outer Join</li> </ul> </li> <li>8. Subqueries <ul style="list-style-type: none"> <li>• With IN clause</li> <li>• With EXISTS clause</li> </ul> </li> <li>9. Views <ul style="list-style-type: none"> <li>• Creating Views (with and without check option)</li> <li>• Dropping views</li> <li>• Selecting from a view</li> </ul> </li> <li>10. DCL statements <ul style="list-style-type: none"> <li>• Granting and revoking permissions</li> </ul> </li> </ol>	
<p><b>USCSP105</b></p>	<p><b>Discrete Mathematics</b></p> <ol style="list-style-type: none"> <li>1. Graphs of standard functions such as absolute value function, inverse function, logarithmic and exponential functions, flooring and ceiling functions, trigonometric functions over suitable intervals.</li> <li>2. Partial ordering sets, Hasse diagram and Lattices.</li> <li>3. Recurrence relation.</li> <li>4. Different counting principles.</li> <li>5. Finite state Automata and Finite state machines.</li> <li>6. Warshall's Algorithm.</li> <li>7. Shortest Path algorithms.</li> <li>8. Operations on graph.</li> <li>9. Breadth and Depth First search algorithms.</li> <li>10. Concept of searching, inserting and deleting from binary search trees.</li> </ol>	
<p><b>USCSP106</b></p>	<p><b>Descriptive Statistics and Introduction to Probability (To be implemented using R)</b></p> <ol style="list-style-type: none"> <li>1. Frequency distribution and data presentation</li> <li>2. Measures of central tendency</li> <li>3. Data entry using, functions, c(), scan (), Creating vectors, Mathematical Operations: ** +/-/*/ / ^ , exp, log, log10, etc, creating vector of text type, useful functions: data, frame, matrix operations, seq(), split() etc.</li> <li>4. Frequency distribution using cut(), table()</li> <li>5. Data presentation</li> <li>6. Summary Statistics (measures of central tendency, dispersion)</li> <li>7. Measures of skewness and kurtosis</li> <li>8. Correlation and regression</li> <li>9. Probability</li> <li>10. Conditional probability</li> </ol>	

## Semester II - Theory

Course: USCS201	Programming with C (Credits : 2 Lectures/Week: 3)	
<p><b>Objectives:</b> The objective of this course is to provide a comprehensive study of the C programming language, stressing upon the strengths of C, which provide the students with the means of writing modular, efficient, maintainable, and portable code.</p> <p><b>Expected Learning Outcomes</b></p> <ol style="list-style-type: none"> <li>1) Students should be able to write, compile and debug programs in C language.</li> <li>2) Students should be able to use different data types in a computer program.</li> <li>3) Students should be able to design programs involving decision structures, loops and functions.</li> <li>4) Students should be able to explain the difference between call by value and call by reference</li> <li>5) Students should be able to understand the dynamics of memory by the use of pointers.</li> <li>6) Students should be able to use different data structures and create/update basic data files.</li> </ol>		
Unit I	<p><b>Structure of C program:</b> Header and body, Use of comments. Interpreters vs compilers, Python vs C. Compilation of a program. Formatted I/O: printf(), scanf().</p> <p><b>Data:</b> Variables, Constants, data types like: int, float char, double and void, short and long size qualifiers, signed and unsigned qualifiers. Compare with datatypes in Python. Compare static typing in C vs dynamic typing in Python</p> <p><b>Variables:</b> Declaring variables, scope of the variables according to block, hierarchy of data types. Compare explicit declarations in C with implicit declarations in Python.</p> <p><b>Types of operators:</b> Arithmetic, relational, logical, compound assignment, increment and decrement, conditional or ternary, bitwise and comma operators. Precedence and order of evaluation, statements and Expressions. Automatic and explicit type conversion.</p> <p><b>Iterations:</b> Control statements for decision making: (i) Branching: if statement, else.. if statement, (does the writer mean if-else or nested ifs)switch statement. (ii) Looping: while loop, do.. while, for loop. (iii) Jump statements: break, continue and goto.</p>	15L
Unit II	<p><b>Arrays:</b> (One and two dimensional), declaring array variables, initialization of arrays, accessing array elements. Compare array types of C with list and tuple types of Python.</p> <p><b>Data Input and Output functions:</b> Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().</p> <p><b>Manipulating Strings:</b> Declaring and initializing String variables, Character and string handling functions. Compare with Python strings.</p> <p><b>Functions:</b> Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values.</p> <p><b>Recursion:</b> Definition, Recursive functions.</p>	15L

Unit III	<p><b>Pointer:</b> Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Using Pointers with Arrays, Using Pointers with Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers.</p> <p><b>Dynamic Memory Allocation:</b> malloc(), calloc(), realloc(), free() and sizeof operator. Compare with automatic garbage collection in Python.</p> <p><b>Structure:</b> Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures. Compare C structures with Python tuples.</p> <p><b>Unions:</b> Defining and working with unions.</p> <p><b>File handling:</b> Different types of files like text and binary, Different types of functions: fopen(), fclose(), fgetc(), fputc(), fgets(), fputs(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek().</p>	15L
<p><b>Text books:</b></p> <ol style="list-style-type: none"> <li>1. Programming in ANSI C (Third Edition) : E Balagurusamy, TMH</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Pradip Dey, Manas Ghosh, "Programming in C", second edition, Oxford University Press</li> <li>2. Yashavant P. Kanetkar. " Let Us C", BPB Publications</li> </ol>		

<b>Course:</b> <b>USCS202</b>	<b>Programming with Python – II</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objective:</b>  The objective of this paper is to explore the style of structured programming to give the idea to the students how programming can be used for designing real-life applications by reading/writing to files, GUI programming, interfacing database/networks and various other features.</p> <p><b>Expected Learning Outcomes</b></p> <ol style="list-style-type: none"> <li>1) Students should be able to understand how to read/write to files using python.</li> <li>2) Students should be able to catch their own errors that happen during execution of programs.</li> <li>3) Students should get an introduction to the concept of pattern matching.</li> <li>4) Students should be made familiar with the concepts of GUI controls and designing GUI applications.</li> <li>5) Students should be able to connect to the database to move the data to/from the application.</li> <li>6) 6)Students should know how to connect to computers, read from URL and send email.</li> </ol>		
Unit I	<p><b>Python File Input-Output:</b> Opening and closing files, various types of file modes, reading and writing to files, manipulating directories.  Iterables, iterators and their problemsolving applications.</p> <p><b>Exception handling:</b> What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise.</p> <p><b>Regular Expressions:</b> Concept of regular expression, various types of regular expressions, using match function.</p>	15 L
Unit II	<p><b>GUI Programming in Python (using Tkinter/wxPython/Qt)</b>  What is GUI, Advantages of GUI, Introduction to GUI library. Layout management, events and bindings, fonts, colours, drawing on canvas (line, oval, rectangle, etc.)  Widgets such as : frame, label, button, checkbutton, entry, listbox, message, radiobutton, text, spinbox etc</p>	15 L
Unit III	<p><b>Database connectivity in Python:</b> Installing mysql connector, accessing connector module module, using connect, cursor, execute &amp; close functions, reading single &amp; multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity.</p> <p><b>Network connectivity:</b> Socket module, creating server-client programs, sending email, reading from URL</p>	15 L
<p><b>Text books:</b></p> <ol style="list-style-type: none"> <li>1. Paul Gries , Jennifer Campbell, Jason Montojo, <i>Practical Programming: An Introduction to Computer Science Using Python 3</i>, Pragmatic Bookshelf, 2/E 2014</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. James Payne , <i>Beginning Python: Using Python 2.6 and Python 3</i>, Wiley India, 2010</li> <li>2. A. Lukaszewski, <i>MySQL for Python: Database Access Made Easy</i>, Pact Publisher, 2010</li> </ol>		

<b>Course:</b> <b>USCS203</b>	<b>Linux</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objectives:</b>  This course introduces various tools and techniques commonly used by Linux programmers, system administrators and end users to achieve their day to day work in Linux environment. It is designed for computer students who have limited or no previous exposure to Linux.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Upon completion of this course, students should have a good working knowledge of Linux, from both a graphical and command line perspective, allowing them to easily use any Linux distribution.</li> <li>2) This course shall help student to learn advanced subjects in computer science practically.</li> <li>3) Student shall be able to progress as a Developer or Linux System Administrator using the acquired skill set.</li> </ol>		
Unit I	<p><b>Introduction</b>  History of Linux, Philosophy, Community, Terminology, Distributions, Linux kernel vs distribution.  Why learn Linux? Importance of Linux in software ecosystem: web servers, supercomputers, mobile, servers.</p> <p><b>Installation</b>  Installation methods, Hands on Installation using CD/DVD or USB drive.</p> <p><b>Linux Structure</b>  Linux Architecture, Filesystem basics, The boot process, init scripts, runlevels, shutdown process, Very basic introductions to Linux processes, Packaging methods: rpm/deb, Graphical Vs Command line.</p>	15L
Unit II	<p><b>Graphical Desktop</b>  Session Management, Basic Desktop Operations, Network Management, Installing and Updating Software, Text editors: gedit, vi, vim, emacs, Graphics editors, Multimedia applications.</p> <p><b>Command Line</b>  Command line mode options, Shells, Basic Commands, General Purpose Utilities, Installing Software, User management, Environment variables, Command aliases.</p> <p><b>Linux Documentation</b>  man pages, GNU info, help command, More documentation sources</p> <p><b>File Operations</b>  Filesystem, Filesystem architecture, File types, File attributes, Working with files, Backup, compression</p>	15L
Unit III	<p><b>Security</b>  Understanding Linux Security, Uses of root, sudo command, working with passwords, Bypassing user authentication, Understanding ssh</p> <p><b>Networking</b>  Basic introduction to Networking, Network protocols: http, ftp etc., IP address, DNS, Browsers, Transferring files.  ssh, telnet, ping, traceroute, route, hostname, networking GUI.</p> <p><b>Basic Shell Scripting</b>  Features and capabilities, Syntax, Constructs, Modifying files, Sed, awk command, File manipulation utilities, Dealing with large files and Text, String manipulation, Boolean expressions, File tests, Case, Debugging, Regular expressions</p>	15L

**Text book:**

- 1) Unix Concepts and Applications by Sumitabha Das.
- 2) Official Ubuntu Book, 8th Edition, by Matthew Helmke & Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, Prentice Hall

**Additional References:**

- 1) Linux kernel Home: <http://kernel.org>
- 2) Open Source Initiative: <https://opensource.org/>
- 3) The Linux Foundation: <http://www.linuxfoundation.org/>

<b>Course:</b> <b>USCS204</b>	<b>Data Structures</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<p><b>Objectives:</b> To explore and understand the concepts of Data Structures and its significance in programming. Provide and holistic approach to design, use and implement abstract data types. Understand the commonly used data structures and various forms of its implementation for different applications using Python.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Learn about Data structures, its types and significance in computing</li> <li>2) Explore about Abstract Data types and its implementation</li> <li>3) Ability to program various applications using different data structure in Python</li> </ol>		
Unit I	<p><b>Abstract Data Types:</b> Introduction, The Date Abstract Data Type, Bags, Iterators. Application</p> <p><b>Arrays:</b> Array Structure, Python List, Two Dimensional Arrays, Matrix Abstract Data Type, Application</p> <p><b>Sets and Maps:</b> Sets-Set ADT, Selecting Data Structure, List based Implementation, Maps-Map ADT, List Based Implementation, Multi-Dimensional Arrays-Multi-Array ADT, Implementing Multiarrays, Application</p> <p><b>Algorithm Analysis:</b> Complexity Analysis-Big-O Notation, Evaluating Python Code, Evaluating Python List, Amortized Cost, Evaluating Set ADT, Application</p> <p><b>Searching and Sorting:</b> Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted List, Maintaining sorted Lists.</p>	15L
Unit II	<p><b>Linked Structures:</b> Introduction, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Bag ADT-Linked List Implementation. Comparing Implementations, Linked List Iterators, More Ways to Build Kinked Lists, Applications-Polynomials</p> <p><b>Stacks:</b> Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-Balanced Delimiters, Evaluating Postfix Expressions</p> <p><b>Queues:</b> Queue ADT, Implementing Queue-Using Python List, Circular Array, Using List, Priority Queues- Priority Queue ADT, Bounded and unbounded Priority Queues</p> <p><b>Advanced Linked List:</b> Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation, Multi Lists</p>	15L



Unit III	<p><b>Recursion:</b> Recursive Functions, Properties of Recursion, Its working, Recursive Applications</p> <p><b>Hash Table:</b> Introduction, Hashing-Linear Probing, Clustering, Rehashing, Separate Chaining, Hash Functions</p> <p><b>Advanced Sorting:</b> Merge Sort, Quick Sort, Radix Sort, Sorting Linked List</p> <p><b>Binary Trees:</b> Tree Structure, Binary Tree-Properties, Implementation and Traversals, Expression Trees, Heaps and Heapsort, Search Trees</p>	15L
<p><b>Text book:</b></p> <ol style="list-style-type: none"> <li>1) <i>Data Structure and algorithm Using Python</i>, Rance D. Necaie, 2016 Wiley India Edition</li> <li>2) <i>Data Structure and Algorithm in Python</i>, Michael T. Goodrich, Robertom Tamassia, M. H. Goldwasser, 2016 Wiley India Edition</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1) <i>Data Structure and Algorithmic Thinking with Python-</i> Narasimha Karumanchi, 2015, Careermonk Publications</li> <li>2) <i>Fundamentals of Python: Data Structures</i>, Kenneth Lambert, Delmar Cengage Learning</li> </ol>		

<b>Course:</b> USCS205	<b>Calculus</b> (Credits : 2 Lectures/Week: 3)	
<p><b>Objectives:</b></p> <p>The course is designed to have a grasp of important concepts of Calculus in a scientific way. It covers topics from as basic as definition of functions to partial derivatives of functions in a gradual and logical way. The learner is expected to solve as many examples as possible to get complete clarity and understanding of the topics covered.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Understanding of Mathematical concepts like limit, continuity, derivative, integration of functions.</li> <li>2) Ability to appreciate real world applications which uses these concepts.</li> <li>3) Skill to formulate a problem through Mathematical modeling and simulation.</li> </ol>		
Unit I	<p><b>DERIVATIVES AND ITS APPLICATIONS:</b></p> <p>Review of Functions, limit of a function, continuity of a function, derivative function. Derivative In Graphing And Applications: Analysis of Functions: Increase, Decrease, Concavity, Relative Extrema; Graphing Polynomials, Rational Functions, Cusps and Vertical Tangents. Absolute Maxima and Minima, Applied Maximum and Minimum Problems, Newton's Method.</p>	15L
Unit II	<p><b>INTEGRATION AND ITS APPLICATIONS:</b></p> <p>An Overview of the Area Problem, Indefinite Integral, Definition of Area as a Limit; Sigma Notation, Definite Integral, Evaluating Definite Integrals by Substitution, Area Between Two Curves, Length of a Plane Curve. Numerical Integration: Simpson's Rule. Modeling with Differential Equations, Separation of Variables, Slope Fields, Euler's Method, First-Order Differential Equations and Applications.</p>	15L
Unit III	<p><b>PARTIAL DERIVATIVES AND ITS APPLICATIONS:</b></p> <p>Functions of Two or More Variables Limits and Continuity Partial Derivatives, Differentiability, Differentials, and Local Linearity, Chain Rule, Directional Derivatives and Gradients, Tangent Planes and Normal, Vectors, Maxima and Minima of Functions of Two Variables.</p>	15L

**Textbook:**

1. Calculus: Early transcendental (10th Edition): Howard Anton, Irl Bivens, Stephen Davis, John Wiley & sons, 2012.

**Additional References:**

1. Calculus and analytic geometry (9th edition): George B Thomas, Ross L Finney, Addison Wesley, 1995
2. Calculus: Early Transcendentals (8th Edition): James Stewart, Brooks Cole, 2015.
3. Calculus (10th Edition): Ron Larson, Bruce H. Edwards, Cengage Learning, 2013.
4. Thomas' Calculus (13th Edition): George B. Thomas, Maurice D. Weir, Joel R. Hass, Pearson, 2014.

<b>Course:</b> <b>USCS206</b>	<b>Statistical Methods and Testing of Hypothesis</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<b>Objectives:</b> The purpose of this course is to familiarize students with basics of Statistics. This will be essential for prospective researchers and professionals to know these basics.		
<b>Expected Learning Outcomes:</b> 1) Enable learners to know descriptive statistical concepts 2) Enable study of probability concept required for Computer learners		
Unit I	<b>Standard distributions:</b> random variable; discrete, continuous, expectation and variance of a random variable, pmf, pdf, cdf, reliability, Introduction and properties without proof for following distributions; binomial, normal, chi-square, t, F. Examples	15L
Unit II	<b>Hypothesis testing:</b> one sided, two sided hypothesis, critical region, p-value, tests based on t, Normal and F, confidence intervals. Analysis of variance : one-way, two-way analysis of variance	15L
Unit III	<b>Non-parametric tests:</b> need of non-parametric tests, sign test, Wilcoxon's signed rank test, run test, Kruskal-Wallis tests. Post-hoc analysis of one-way analysis of variance : Duncan's test Chi-square test of association	15L
<b>Text Book:</b> 1. Trivedi, K.S.(2009) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi		
<b>Additional References:</b> 1. Ross, S.M. (2006): A First course in probability. 6 <sup>th</sup> Ed <sup>n</sup> Pearson 2. Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): Common statistical tests. Satyajeet Prakashan, Pune 3. Gupta, S.C. and Kapoor, V.K. (2002) : Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi 4. Gupta, S.C. and Kapoor, V.K. (4 <sup>th</sup> Edition) : Applied Statistics, S. Chand and Son's, New Delhi 5. Montgomery, D.C. (2001): Planning and Analysis of Experiments, Wiley.		

<b>Course:</b> <b>USCS207</b>	<b>Green Technologies</b> <b>(Credits : 2 Lectures/Week: 3)</b>	
<b>Objectives:</b> To familiarize with the concept of Green Computing and Green IT infrastructure for making computing and information system environment sustainable. Encouraging optimized software and hardware designs for development of Green IT Storage, Communication and Services. To highlight useful approaches to embrace green IT initiatives.		

<p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Learn about green IT can be achieved in and by hardware, software, network communication and data center operations.</li> <li>2) Understand the strategies, frameworks, processes and management of green IT</li> </ol>		
Unit I	<p><b>Green IT Overview:</b> Introduction , Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Green I , Holistic Approach to Greening IT, Greening IT, Applying IT for Enhancing Environmental Sustainability, Green IT Standards and Eco-Labeling of IT , Enterprise Green IT Strategy, Green Washing, Green IT: Burden or Opportunity?</p> <p><b>Green Devices and Hardware:</b> Introduction , Life Cycle of a Device or Hardware, Reuse, Recycle and Dispose</p> <p><b>Green Software:</b> Introduction , Processor Power States , Energy-Saving Software Techniques, Evaluating and Measuring Software Impact to Platform Power</p> <p><b>Sustainable Software Development:</b> Introduction, Current Practices, Sustainable Software, Software Sustainability Attributes, Software Sustainability Metrics, Sustainable Software Methodology, Defining Actions</p>	15L
Unit II	<p><b>Green Data Centres:</b> Data Centres and Associated Energy Challenges, Data Centre IT Infrastructure, Data Centre Facility Infrastructure: Implications for Energy Efficiency, IT Infrastructure Management, Green Data Centre Metrics</p> <p><b>Green Data Storage:</b> Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management</p> <p><b>Green Networks and Communications:</b> Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards</p> <p><b>Enterprise Green IT Strategy:</b> Introduction, Approaching Green IT Strategies, Business Drivers of Green IT Strategy, Business Dimensions for Green IT Transformation, Organizational Considerations in a Green IT Strategy, Steps in Developing a Green IT Strategy, Metrics and Measurements in Green Strategies.</p>	15L
Unit III	<p><b>Sustainable Information Systems and Green Metrics:</b> Introduction, Multilevel Sustainable Information, Sustainability Hierarchy Models, Product Level Information, Individual Level Information, Functional Level Information, Organizational Level Information, Measuring the Maturity of Sustainable ICT</p> <p><b>Enterprise Green IT Readiness:</b> Introduction, Readiness and Capability, Development of the G-Readiness Framework, Measuring an Organization's G-Readiness</p> <p><b>Sustainable IT Services: Creating a Framework for Service Innovation:</b> Introduction, Factors Driving the Development of Sustainable IT, Sustainable IT Services (SITS), SITS Strategic Framework</p> <p><b>Green Enterprises and the Role of IT:</b> Introduction, Organizational and Enterprise Greening, Information Systems in Greening Enterprises, Greening the Enterprise: IT Usage and Hardware, Inter-organizational Enterprise Activities and Green Issues</p>	15L
<p><b>Text book:</b></p> <ol style="list-style-type: none"> <li>1) <i>Harnessing Green IT: Principles and Practices</i>, San Murugesan, G. R. Ganadharan, Wiley &amp; IEEE.</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1) <i>Green IT</i>, Deepak Shikarpur, Vishwkarma Publications, 2014</li> <li>2) <i>Green Communications: Principles, Concepts and Practice</i>- Samdanis et al, J. Wiley</li> <li>3) <i>Green IT for Sustainable Business Practice: An ISEB Foundation Guide</i>, Mark G. O'Neill, The Chartered Institute for IT, 2010</li> </ol>		

## Semester II – Practical

---

Course: USCSP2	Practical of USCS201 + USCS202 + USCS203+USCS204+ USCS205+ USCS206 (Credits : 6, Lectures/Week: 18)	
USCSP201	<b>Programming with C</b> <ol style="list-style-type: none"><li>1. Programs to understand the basic data types and I/O.</li><li>2. Programs on Operators and Expressions</li><li>3. Programs on decision statements.</li><li>4. Programs on looping.</li><li>5. Programs on arrays.</li><li>6. Programs on functions.</li><li>7. Programs on structures and unions.</li><li>8. Programs on pointers.</li><li>9. Programs on string manipulations.</li><li>10. Programs on basic file operations.</li></ol>	
USCSP202	<b>Programming with Python-II</b> <ol style="list-style-type: none"><li>1. Programs to read and write files.</li><li>2. Programs with iterables and iterators.</li><li>3. Program to demonstrate exception handling.</li><li>4. Program to demonstrate the use of regular expressions.</li><li>5. Program to show draw shapes &amp; GUI controls.</li><li>6. Program to create server-client and exchange basic information.</li><li>7. Program to send email &amp; read contents of URL.</li></ol>	

<b>USCSP203</b>	<p><b>Linux</b></p> <ol style="list-style-type: none"> <li>1. Linux Installation:       <ol style="list-style-type: none"> <li>a. Install your choice of Linux distribution e.g. Ubuntu, Fedora, Debian.</li> <li>b. Try different installation media like CD/DVD, USB Drive to install.</li> <li>c. Customize desktop environment by changing different default options like changing default background, themes, screensavers.</li> </ol> </li> <li>2.       <ol style="list-style-type: none"> <li>a. Screen Resolution: Ascertain the current screen resolution for your desktop.</li> <li>b. Networking: Get the current networking configuration for your desktop. Are you on a wired or a wireless connection? What wireless networks are available, if any?</li> <li>c. Time Settings Change the time zone of your system to (or New York Time if you are currently in Indian time). How does the displayed time change? After noting the time change, change the time zone back to your local time zone.</li> </ol> </li> <li>3. Installing and Removing Software:       <ol style="list-style-type: none"> <li>a. Install gcc package. Verify that it runs, and then remove it.</li> </ol> </li> <li>4. Documentations:       <ol style="list-style-type: none"> <li>a. Finding Info Documentation: From the command line: bring up the info page for the grep command. Bring up the usage section.</li> <li>b. Finding man pages From the command line: Bring up the man page for the 'ls' command. Scroll down to the EXAMPLES section.</li> <li>c. Finding man pages by Topic What man pages are available that document file compression?</li> <li>d. Finding man pages by Section From the command line, bring up the man page for the printf library function. Which manual page section are library functions found?</li> <li>e. Command-Line Help List the available options for the mkdir command. How can you do this?</li> </ol> </li> <li>5. Command line operations:       <ol style="list-style-type: none"> <li>a. Install any newpackage on your system</li> <li>b. Remove the package installed</li> <li>c. Find the passwd file in / using find command</li> <li>d. Create a symbolic link to the file you found in last step</li> <li>e. Create an empty file example.txt and move it in /tmp directory using relative pathname.</li> <li>f. Delete the file moved to /tmp in previous step using absolute path.</li> <li>g. Find the location of ls, ps, bash commands.</li> </ol> </li> <li>6. File Operations:       <ol style="list-style-type: none"> <li>a. Explore mounted filesystems on your system.</li> <li>b. What are different ways of exploring mounted filesystems on Linux?</li> <li>c. Archive and backup your home directory or work directory using tar, gzip commands.</li> <li>d. Use dd command to create files and explore different options to dd.</li> <li>e. Use diff command to create diff of two files.</li> <li>f. Use patch command to patch a file. And analyze the patch using diff command again.</li> </ol> </li> </ol>	
-----------------	--	--

	<ol style="list-style-type: none"> <li>7. Use environment       <ol style="list-style-type: none"> <li>a. Which account are you logged in? How do you find out?</li> <li>b. Display <code>/etc/shadow</code> file using <code>cat</code> and understand the importance of shadow file. How it's different than <code>passwd</code> file.</li> <li>c. Get you current working directory.</li> <li>d. Explore different ways of getting command history, how to run previously executed command without typing it?</li> <li>e. Create alias to most commonly used commands like.</li> </ol> </li> <li>8. Linux Editors: <code>vim/emacs</code> <ol style="list-style-type: none"> <li>a. Create,modify, search, navigate a file in editor.</li> <li>b. Learn all essential commands like search, search/replace, highlight, show line numbers.</li> </ol> </li> <li>9. Linux Security:       <ol style="list-style-type: none"> <li>a. Use of <code>sudo</code> to change user privileges to root</li> <li>b. Identify all operations that require <code>sudo</code> privileges</li> <li>c. Create a new user and add it to <code>sudo</code> configuration file.</li> <li>d. Set password for new user.</li> <li>e. Modify the expiration date for new user using password ageing.</li> <li>f. Delete newly added user.</li> </ol> </li> <li>10. Network:       <ol style="list-style-type: none"> <li>a. Get IP address of your machine using <code>ifconfig</code>.</li> <li>b. If IP is not set, then assign an IP address according to your network settings.</li> <li>c. Get hostname of your machine.</li> <li>d. Use <code>ping</code> to check the network connectivity to remote machines.</li> <li>e. Use <code>telnet/ssh</code> to connect to remote machines and learn the difference between the two.</li> <li>f. Troubleshooting network using <code>traceroute</code>, <code>ping</code>, <code>route</code> commands.</li> </ol> </li> <li>11. Shell Scripting       <ol style="list-style-type: none"> <li>a. Searching with <code>grep</code>: Search for your username in the <code>/etc/passwd</code> file.</li> <li>b. Parsing files with <code>awk</code>: Display in a column a unique list of all the shells used for users in <code>/etc/passwd</code>. Which field in <code>/etc/passwd</code> holds the shell (user command interpreter in the manual page)? How do you make a list of unique entries, that is, no repeated entries?</li> <li>c. Searching and substituting with <code>sed</code>: Search all instances of the user command interpreter (shell) equal to <code>/bin/false</code> in <code>/etc/passwd</code> and substitute with <code>/bin/bash</code> using <code>sed</code>.</li> <li>d. Exit status: write a script which does <code>ls</code> to a non existent file. Display an exit status of the previous command. Now create the file and again display the exit status. In each task send the <code>ls</code> output to <code>/dev/null</code></li> <li>e. Working with files: Write a shell script which will ask user for a directory, create that directory and switch to it and tell the user where you are using <code>pwd</code> command. Now use <code>touch</code> to create some new files followed by displaying the filenames.</li> </ol> </li> </ol>	
--	---	--

	<p>f. Environment variables: Write a script which displays all environment variables on the system.</p> <p>g. Functions: Write a script that asks user for a number (1,2 or 3) which is used to call a function with the number in its name. The function then displays a message with the function number within it, example: "This message is from function number 4."</p> <p>h. Arithmetic: Write a script which will work as arithmetic calculator to add, subtract, multiply, divide. The user should pass an argument on the command line a letter (a,s,m or d) and two numbers. If wrong number of arguments are passed then display an error message. Make use of functions to perform operations.</p> <p>i. Case Statements: Write a script that will be given a month number as the argument and will translate this number into a month name. The result will be printed to stdout.</p> <p>j. Script Arguments and Usage Information: Write a script that takes exactly one argument, a directory name. The script should print that argument back to standard output. Make sure the script generates a usage message if needed and that it handles errors with a message.</p> <p>k. Randomness: Create a script that takes a word as an argument from the user, then appends a random number to the word and display it to the user. Put in a check to make sure the user passed in a word, displaying a usage statement if a word was not passed as an argument.</p> <p>l. Strings: Write a script that will read two strings from the user. The script will perform three operations on the two strings: (1) Use the test command to see if one of the strings is of zero length and if the other is of non-zero length, telling the user of both results. (2) Determine the length of each string and tell the user which is longer or if they are of equal length. (3) Compare the strings to see if they are the same. Let the user know the result.</p> <p>12. Processes</p> <p>a. Background and Foreground Jobs: Create a job that writes the date to an output file thrice, with a gap of 60 seconds and 180 seconds. Check whether the job is running and bring it to foreground job. Stop the foreground job and make it run in the background. Finally, kill the background job and verify its status.</p> <p>b. Scheduling a One-Time Backup: Create job using at to back up files in one directory to another 10 minutes from now.</p> <p>c. Scheduling Repeated Backups: Set up a cron job to backup the files in one directory to another every day at 10 am. Put the commands in file called mycron.</p>	
--	---	--

<p><b>USCSP204</b></p>	<p><b>Data structures</b></p> <ol style="list-style-type: none"> <li>1) Implement Linear Search to find an item in a list.</li> <li>2) Implement binary search to find an item in an ordered list.</li> <li>3) Implement Sorting Algorithms <ol style="list-style-type: none"> <li>a. Bubble sort</li> <li>b. Insertion sort</li> <li>c. Quick sort</li> <li>d. Merge Sort</li> </ol> </li> <li>4) Implement use of Sets and various operations on Sets.</li> <li>5) Implement working of Stacks. (pop method to take the last item added off the stack and a push method to add an item to the stack)</li> <li>6) Implement Program for <ol style="list-style-type: none"> <li>a. Infix to Postfix conversion</li> <li>b. Postfix Evaluation</li> </ol> </li> <li>7) Implement the following <ol style="list-style-type: none"> <li>a. A queue as a list which you add and delete items from.</li> <li>b. A circular queue. (The beginning items of the queue can be reused).</li> </ol> </li> <li>8) Implement Linked list and demonstrate the functionality to add and delete items in the linked list.</li> <li>9) Implement Binary Tree and its traversals.</li> <li>10) Recursive implementation of <ol style="list-style-type: none"> <li>a. Factorial</li> <li>b. Fibonacci</li> <li>c. Tower of Hanoi</li> </ol> </li> </ol>	
<p><b>USCSP205</b></p>	<p><b>Calculus</b></p> <ol style="list-style-type: none"> <li>1. Continuity of functions; Derivative of functions</li> <li>2. Increasing, decreasing, concave up and concave down functions</li> <li>3. Relative maxima, relative minima, absolute maxima, absolute minima</li> <li>4. Newton's method to find approximate solution of an equation</li> <li>5. Area as a limit and length of a plane curve</li> <li>6. Numerical integration using Simpson's rule</li> <li>7. Solution of a first order first degree differential equation, Euler's method</li> <li>8. Calculation of Partial derivatives of functions</li> <li>9. Local linear approximation and directional derivatives</li> <li>10. Maxima and minima of functions of two variables</li> </ol>	
<p><b>USCSP206</b></p>	<p><b>Statistical Methods and Testing of Hypothesis</b></p> <ol style="list-style-type: none"> <li>1. Problems based on binomial distribution</li> <li>2. Problems based on normal distribution</li> <li>3. Property plotting of binomial distribution</li> <li>4. Property plotting of normal distribution</li> <li>5. Plotting pdf, cdf, pmf, for discrete and continuous distribution</li> <li>6. t test, normal test, F test</li> <li>7. Analysis of Variance</li> <li>8. Non parametric tests- I</li> <li>9. Non- Parametric tests – II</li> <li>10. Post-hoc analysis of one-way analysis</li> </ol>	



## Evaluation Scheme

### I. Internal Exam-25 Marks

#### (i) Test– 20 Marks

20 marks Test – Duration 40 mins

It will be conducted either using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment) Or a test based on an equivalent online course on the contents of the concerned course(subject) offered by or build using MOOC (Massive Open Online Course) platform.

#### (ii) 5 Marks - Active participation in routine class instructional deliveries

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

### II. External Examination- 75 Marks

#### (i) Duration - 2.5 Hours.

#### (ii) Theory question paper pattern:-

All questions are compulsory.		
Question	Based on	Marks
Q.1	Unit I	20
Q.2	Unit II	20
Q.3	Unit III	20
Q.4	Unit I,II and III	15

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be sub-divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

### III. Practical Examination – 300 marks (50 marks x 6 core papers)

- Each core subject carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75 % practical from each core subjects are required to be completed and written in the journal.

**(Certified Journal is compulsory for appearing at the time of Practical Exam)**

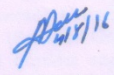
-----

**UNIVERSITY OF MUMBAI**  
**No. UG/42 of 2016-17**

**CIRCULAR:-**

A reference is invited to the Syllabi relating to the B.Sc. degree course , **vide** this office Circular No. UG/231 of 2009, dated 16<sup>th</sup> June, 2009 and the Principals of affiliated Colleges in Science are hereby informed that the recommendation made by Ad-hoc-Board of Studies in Science at its meeting held on 20<sup>th</sup> May, 2016 has been accepted by the Academic Council meeting held on 23<sup>rd</sup> May, 2015 **vide** item No. 4.12 and that in accordance therewith, the revised syllabus as per the Credit Based Semester and Grading System for S.Y. B.Sc. Computer Science (Sem.III & IV), which are available on the University's web site ([www.mu.ac.in](http://www.mu.ac.in)) and that the same has been brought into force with effect from the academic year 2016-17.

MUMBAI – 400 032  
5<sup>th</sup> August, 2016

  
(Dr.M.A.Khan)  
REGISTRAR

To,

The Principals of the affiliated Colleges in Science and the Heads of Recognized Institutions concerned.

**A.C/4.12/23.05.2016**

\*\*\*\*\*

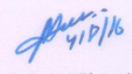
No. UG/42-A of 2016

MUMBAI-400 032

5<sup>th</sup> August, 2016

Copy forwarded with Compliments for information to:-

- 1) The Deans, faculties of Science,
- 2) The Chairman, Board of Studies in Science,
- 3) The Professor-cum-Director, Institute of Distance & Open Learning (IDOL)
- 4) The Director, Board of College and University Development,
- 5) The Co-Ordinator, University Computerization Centre,
- 6) The Controller of Examinations.

  
(Dr.M.A.Khan)  
REGISTRAR

PTO..

# **UNIVERSITY OF MUMBAI**



**Syllabus for the S.Y.B.Sc.**

**Program: B.Sc.**

**Course: Computer Science**

(Credit Based Semester and Grading System with  
effect from the academic year 2016–2017)

# Preamble

With rapid and steady advances in diverse areas of computing, industry requirements are changing. The three-year B.Sc. Computer Science course is aimed at laying a foundation of software and hardware concepts, supplemented by practical techniques.

The syllabus is designed in such a way that the first year of the course provides core basic concepts of the subject and forms the foundation for further semesters. It attempts to provide technology-oriented students with the knowledge and ability to develop creative solutions, and better understand the effects of future developments of computer technologies.

This syllabus is the continuation to the previous semester's syllabus. It is believed that the syllabus will offer an enriched learning experience and by end of second year, the student will be able to work on several of the prevalent programming /platforms.

**S.Y.B.Sc. Computer Science Syllabus**  
**Credit Based System and Grading System**  
**Academic year 2016-2017**

**SEMESTER III**

<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>Lecture/ Week</b>
USCS301	Discrete Mathematics	2	3
USCS302	Object Oriented Design using UML and Python	2	3
USCS303	Data Structures and Algorithms Using Python	2	3
USCSP03	Practical of USCS301 + USCS302 + USCS303	3	9

**SEMESTER IV**

<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>Lecture/ Week</b>
USCS401	Operating System and Linux	2	3
USCS402	Java Programming	2	3
USCS403	Web Technologies	2	3
USCSP04	Practical of USCS401 + USCS402 + USCS403	3	9

## Semester III – Theory

<b>Course:</b> USCS301	<b>TOPICS (Credits : 02 Lectures/Week:03)</b> <b>Discrete Mathematics</b>	
<p><b>Objectives :</b> To provide an insight into concepts of discrete mathematics and establish its significance in several areas of computational theories.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) To provide overview of theory of discrete objects, starting with relations and partially ordered sets.</li> <li>2) Study about recurrence relations, generating function and operations on them.</li> <li>3) Give an understanding of graphs and trees, which are widely used in software.</li> <li>4) Provide basic knowledge about models of automata theory and the corresponding formal languages.</li> </ol>		
<b>Unit I</b>	<p><b>UNIT I: Relations</b></p> <p>Relations: Definitions and examples. Properties of relations, Partial Ordering sets, Linear ordering, Hasse Daigrams, Maximum and Minimum elements,</p> <p>Recurrence Relation: Definition of recurrence relations,Formulating recurrence relations, Solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients. Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root. Particular solutions of non linear homogeneous recurrence relation, Solution of recurrence relation by the method of generating functions, Applications: Formulate and solve recurrence relations for Fibonacci numbers, Tower of Hanoi.</p>	<b>15 L</b>
<b>Unit II</b>	<p>Graphs, Trees, Counting</p> <p>(a) Graphs : Definition and elementary results, Adjacency matrix, path matrix, Representing relations using diagraphs. Warshall's algorithm- shortest path , Linked representation of a graph, Operations on graph with algorithms - searching in a graph; Insertion in a graph, Deleting from a graph,</p> <p>(b) Trees: Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees, traversing binary trees, binary search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree. (See related topics in Unit 3 of USCS303 Data Structures and Algorithms.)</p> <p>(c) Permutations and Combinations: Partition and Distribution of objects, Permutations and combinations with distinct and indistinct objects, Binomial numbers, Pascal Identity, Vandermonde's Identity, binomial theorem, binomial coefficients and Pascal's triangle.</p>	<b>15 L</b>

<b>Unit III</b>	<p>Languages and Automata:</p> <p>Languages, Grammars and Machines: Chomsky hierarchy of type-0, type-1, type-2 and type-3 grammars; and the languages they generate. Regular expressions and finite state machines, context-free languages and pushdown automata, brief mention of context-sensitive languages and linear bounded automata, recursively enumerable languages and Turing machines. Universal Turing machine and Turing completeness.</p>	<b>15 L</b>
-----------------	--	-------------

**Textbooks:**

1. *Elements of Discrete Mathematics*: C.L. Liu , Tata McGraw- Hill Edition .
2. *Discrete Mathematics and its applications*: Kenneth H. Rosen, Third Edition, McGraw- Hill Inc.
3. *Discrete Mathematics*: Y. N Singh, Wiley India
4. *Discrete Mathematics*: Semyour Lipschutz, Marc Lipson, Schaum’s out lines, McGraw- Hill Inc.

**References:**

1. *Discrete Mathematics*: Norman L. Biggs:, Revised Edition, Clarendon Press Oxford 1989.
2. *Concrete Mathematics*: Graham, Knuth, Patashnik Second Edition, Pearson Education.

<b>Course: USCS302</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03) Object Oriented Design using UML and Python</b>	
<p>Objectives: To clearly understand the concepts of object oriented analysis and design and its application in developing software for real world applications. Introduce Unified Modeling Language and explain its importance in the software development life cycle. Concretely understand the concepts with practical implementations in modern programming environment.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Gain knowledge about principles, components and structure of object oriented programming methodology.</li> <li>2) Understand the behavior and interaction between the software modules through object oriented design.</li> <li>3) Study about various representations, behavioral and architectural modeling techniques.</li> <li>4) Explore use of UML in Object Oriented design and implementation of software designs using Python.</li> </ol> <p>(It is understood that implementation of case studies will not be complete by semester-end. What is more important is, students will appreciate the difficulties faced in even relatively simple analysis and design.)</p> <p><b>Case study for Units I and II</b></p> <p>For practice with UML models and Python coding: Bank branch, with classes - teller, supervisor, cashier, manager (subclasses of the class BankEmployee). Together they daily serve hundreds of customers who visit the bank branch to perform various transactions.</p> <p>Retail banking becomes an uncomplicated exemplar of OO programming because everyone is familiar</p>		

with it. UML constructs (like sequence diagram, class diagram, use cases) are illustrated using the banking case. The implementation or coding is done in Python, in both lecture classes and labs.

For programming purposes, the bank workers have “objects” as their images on the computer network. Each human operates the object that is their image in the computer, from login to timeout and logout.

A customer initiates a transaction by filling in a blank (paper or e-) voucher and handing it to a teller. Once initiated, a transaction remains pending until it is ended/finished (always by a bank worker).

The teller processes the transaction and if needed, forwards it to the cashier and/or supervisor. As a mark of a pending transaction, a token is issued to the customer, and surrendered to the bank at the end of transaction.

The transactions initiated by the customers are: 1. deposit cash, 2. deposit cheques, 3. withdraw cash (with a cheque or voucher), 4. request account statement. (These form 4 high-level UML use cases.)

A teller accepts the voucher; if paper voucher, enters details into e-voucher; checks what kind of transaction (cash/cheque deposit, cash/DD withdrawal); if cheque deposit, send for clearing; if cash deposit, send to cashier; if withdrawal, verify signature cheque or voucher, verify account balance and forward to supervisor; if withdrawal via DD, initiate the DD issue process, print out the DD, send it to supervisor for signature. Further action by the teller on pending vouchers depends on responses from the supervisor re sent vouchers.

A cashier hands out or receives cash as indicated on transaction vouchers sent by the teller or supervisor. An important part of a bank branch is handling sums of cash, receiving and disbursing it, accounting for it, storing it. A cashier handles cash transactions (deposits and withdrawals), maintains accounts of cash on hand.

A supervisor verifies the correctness of the transaction vouchers and updates accounts. The vouchers are filed away, or returned to the teller, or forwarded to the cashier, or cancelled if invalid.

A manager is the final authority and arbiter in the bank branch, a responsible position with cash involved.

### **Case Study for Unit II and III: Simplified Order Processing System**

The Simplified Order Processing System is a system designed for managing how customers place an order, doing payments after receiving the invoice and the ordered products, the retailer should also verify the availability of the stock.

A retailer checks for the availability of goods in the store. If the stock of goods is less than the reorder level, the retailer places an order for goods. The supplier supplies the goods to the store in the system. Once the ordered goods are received at the store, the retailer then arrange them by product or by price, then retailer makes payment. If the stock of goods is available then he will arrange goods for sale.

The retailer then sells the goods directly to the customer. The customer buys the items from retailer. The retailer prepares the bill for all the goods purchased by the customer, then he receives amount either by credit or by cash from customer soon after the product is delivered to the customer. We will not consider customer returns. The overall system is used to manage the goods in the store and does the sales.

The system should comprise of the following set of classes – products, customer, bank, account, order-details, invoice, shipments, etc.



### Case study for Unit III: Employee payroll system for the Bank

Build a new payroll system to allow employees to record timecard information electronically and automatically generate paychecks based on the no. of hours worked and the total sales for commissioned employees. It should provide a desktop interface to allow employee to enter timecard information. Some employees work by the hour, and they are paid hourly rate. They submit timecards that record the date & no. of hours worked for a particular charge number. If someone works for more than 8 hours, they should be paid 1.5 times their normal rate for those extra hours. Hourly workers are paid every friday.

Some employees are paid a flat salary but still they should provide their timecards that record the date and hours worked. They are paid on the last working day of the month.

The employee should be able to query the system for no. of hours, days worked, totals of all hours billed for specific task/charge, total amount received till a given date.

<b>Unit I</b>	<p>Imperative vs object-oriented programming; fundamental ideas of OOP: encapsulation, inheritance, abstraction, polymorphism.</p> <p>Python and UML topics given unitwise are to be taught not serially, but together in an integrated manner.</p> <p><b>UML:</b> Use simple examples first, and the case study next, to develop ideas behind class diagrams, actors and use cases, and use case diagrams. Sequence diagrams to capture use cases. CRC cards (class-responsibility-collaboration). Association, dependency, composition. Inheritance and generalization. Activity diagrams, fork, join. Overview of UML, need for a development process for using UML, the Unified Process.</p> <p><b>Python:</b> The <code>__init__</code> method to initialize newly-created class instances; simple class definitions with and without <code>__init__</code>. Implement classes from the case study: use inheritance for various subclasses of employees, and begin to develop code to implement the sequence diagrams.</p>	<b>15L</b>
<b>Unit II</b>	<p><b>UML:</b> Communication diagram – use simple example to show various components involved (such as classes at generic level of communication or collaboration, objects at instance level of communication; messages that are exchange among different objects, and the order of exchanging the messages among the objects, flow of control, the guard conditions, time at which an object is created and destroyed). Activity diagrams – initial and final states, activity and action states, guard conditions, forking, joining and swimlanes, state transitions based on the outcome of guard conditions.</p> <p><b>Python:</b> create objects by instantiation, implement <code>__del__</code> method to destroy the objects. Based on the conditions that are tested, implement the user-defined method calls along with the necessary parameters to implement the communication among different objects. Multi-threading in order to create multiple flows of control especially when dealing with</p>	<b>15L</b>

	forking and joining. Completion of the coding for bank branch case study, initiation and development of a case study for simplified order processing.	
<b>Unit III</b>	<p><b>UML:</b> State change diagrams, Events and signals, State machines, processes and Threads, time and space, state transitions, initial and final state of an object, sub states.</p> <p>Package diagrams – package and elements that are organized in it such as class diagrams, use case diagrams, etc.</p> <p><b>Python:</b> creation of modules, packages, and importing of packages. Implementation of the order processing case study by creating multiple threads to implement concurrent flows of control, modules and packages.</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) <i>Object Oriented Modeling and Design with UML</i>, Michel Blaha, James Rambaug, Second Edition, Pearson</li> <li>2) <i>Python 3 Object Oriented Programming</i>, Dusty Phillips, PACKT Publishing</li> </ol> <p><b>Reference(s):</b></p> <ol style="list-style-type: none"> <li>3) <i>Object-Oriented analysis and Design: Understanding system Development with UML 2.0</i>, Mike O'Docherty, Wiley India</li> <li>4) <i>Object Oriented Analysis and Design with UML</i>, I.K International Publishing</li> <li>5) <i>Object Oriented Design and Patterns</i> - Cay Hortsman, Wiley India</li> <li>6) <i>Introduction to Computer Science using Python</i>, Charles Diebrach,</li> </ol>		

<b>Course:</b> <b>USCS303</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b> <b>Data Structures and Algorithms Using Python</b>	
<p><b>Objective:</b> The objective of this course is to make the learner understand the basic concepts and algorithms of different data structures and its applications using Python.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Students should be able to understand the meaning of data structures and its different types.</li> <li>2) Students should be able to understand the algorithm of different data structures before implementing it using Python</li> <li>3) Students should be able to develop the logic for implementing data structures.</li> </ol>		
<b>Unit I</b>	<p><b>Algorithm analysis</b>  Problem, size of problem (symbol <math>n</math>); runtime resources time <math>T(n)</math>, space <math>S(n)</math>; worst case, best case, average case. Measuring running time as a function of <math>n</math> with the wall clock (using function <code>time()</code> of module <code>time</code>); advantages and disadvantages.</p> <p><b>7 standard functions:</b> constant <math>c</math>, <math>\log n</math>, <math>n</math>, <math>n \log n</math>, <math>n^2</math>, <math>n^3</math>, exponential <math>c^n</math> or <math>2^n</math>; growth of these functions as <math>n</math> grows: for constants <math>c_1</math> and <math>c_2</math>, compare <math>c_1 * f_1(n)</math> with <math>c_2 * f_2(n)</math> (for functions <math>f_1</math> and <math>f_2</math> from the set of these 7 functions); conclude that one function grows faster than another independent of the values of the constants.</p> <p><b>Operation count, unit steps (constant time):</b> arithmetic operation (or expression evaluation or assignment), comparison (with Boolean operators <math>&lt;</math>, <math>==</math>, <math>&gt;</math>) function call and/or function return, element access (for compound types); can even treat a single loop iteration as a constant-time unit step.</p> <p><b>Asymptotic analysis:</b> upper bounds with <math>A</math> (at most) and <math>O</math> notation; lower bounds with <math>\Omega</math> notation, upper and lower bounds with <math>\Theta</math> notation.</p> <p><b>Problem-solving methods:</b> greedy method, divide-and-conquer, dynamic programming (briefly, during all 3 units)</p> <p><b>Abstract data types</b> (with associated operations and applications) Define the ADTs as Python classes, and their operations as class methods.</p> <p><b>(i) stacks:</b> operations <code>push()</code>, <code>pop()</code>, <code>is_empty()</code>; <code>stacktop()</code>, <code>len()</code> implementation using lists; applications: reverse a sequence, match parentheses in an expression (or html tags); evaluate a postfix expression.</p>	15 L
<b>Unit II</b>	<p><b>(ii) queues:</b> operations <code>enqueue ()</code> and <code>dequeue()</code>, i.e., <code>enter()</code> and <code>exit()</code>, <code>is_empty()</code>, <code>first()</code>, <code>last()</code>; implementation using Python lists; applications: simulation of a single-window queue (uniform, Gaussian and other distributions are available in the Python module <code>random</code>).</p> <p><b>(iii) Singly, doubly and circularly linked lists</b>, with head and optional tail; implementation of list nodes as Python objects; operations: insertion and deletion at the front and the rear of the list, search for a value in a list, delete a value in a list; applications: simulate stack and queue, maintain a set of data in sorted order. Linear search in linked lists.</p> <p><b>(iv) trees and binary trees</b>, definitions and properties; insertion and deletion of a tree node</p>	15 L

<b>Unit III</b>	<p>(v) <b>trees and binary trees</b>, implementation of binary trees in lists and in linked structures; applications: preorder, inorder and postorder traversals of binary trees; binary search trees; breadth-first and depth-first tree traversals.</p> <p>(vi) <b>graphs</b>: directed and undirected graphs; implementation using adjacency matrix and adjacency list; graph traversal algorithms: depth first and breadth first traversals, application: shortest paths</p> <p>(vii) <b>map ADT</b>, Python classes dictionary and set; applications.</p>	15 L
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) <i>Data Structures and Algorithms in Python</i>, Goodrich, Tamassia, Goldwasser, 2016 J. Wiley</li> <li>2) <i>Data Structures and Algorithms Using Python</i> - Rance D. Necaie, College of William and Mary, 2016, J. Wiley</li> </ol> <p><b>Reference(s)</b></p> <ol style="list-style-type: none"> <li>1) <i>Data Structure and Algorithmic Thinking with Python-</i> Narasimha Karumanchi, 2015, Careermonk Publications</li> <li>2) <i>Fundamentals of Python: Data Structures</i>, Kenneth Lambert, Delmar Cengage Learning</li> </ol>		

### Semester III Practical

<b>USCSP03</b>	<p><b>Practical/Tutorial of USCS301 + USCS302 + USCS303</b> (Credits: 03, Practical/Week: 09)</p>
	<p><b>Discrete Mathematics :</b></p> <ol style="list-style-type: none"> <li>1) Problems based on Boolean Algebra</li> <li>2) Problems based on Sets and draw Venn diagram</li> <li>3) Problems based on Relations</li> <li>4) Use Propositional Logic for representing and solving problems.</li> <li>5) Use Predicate logic and WFF for representing and solving problems</li> <li>6) Problems based on Recurrence relations</li> <li>7) Problems based on generating functions.</li> <li>8) Problems based on Permutation theory</li> <li>9) Problems based on Combination theory</li> <li>10) Problems based on Graphs</li> </ol>
	<p><b>Object Oriented Design using UML and Python :</b></p> <ol style="list-style-type: none"> <li>1. Draw class diagram for the Unit 1 case study (specify the classes, i.e., name, attributes and methods but do not draw the edges between the classes). Start the corresponding Python class definitions (state the stubs for attributes and methods but do not write the code).</li> <li>2. Develop the important use cases for the case study (specify the actors, the use case initiators and write the use case descriptions). Complete the class</li> </ol>

	<p>diagram by drawing the associations.</p> <p>3. Develop the sequence diagram for a few important scenarios of the case study. Fill in the corresponding Python code.</p> <p>4-5. Develop the code for the bank branch case study.</p> <p>6-9. UML diagrams – Use case , class, sequence, package diagrams and develop Python code for the order processing case study.</p> <p>10-11. Create multiple threads, provide synchronization for the order processing case study using Python</p> <p>12-16. Draw use case, class, activity, communication and sequence diagrams for employee payroll case study and also create a payroll package with all the necessary classes for the case study and implement the methods of it by calling them in a module called employee_payroll.py</p>
	<p><b>Data Structures and Algorithms Using Python :</b></p> <ol style="list-style-type: none"> <li>1) Find out space and time complexity for a given non recursive program.</li> <li>2) Find out space and time complexity for a given recursive (like Fibonacci) program.</li> <li>3) Write a program to implement stack and its applications.</li> <li>4) Write a program to implement queue and its applications.</li> <li>5) Write a program to implement singly linked list and its applications.</li> <li>6) Write a program to implement doubly linked list and its applications.</li> <li>7) Write a program to perform insertion and deletion of a node from a tree.</li> <li>8) Write a program to print pre-order, post-order and in-order traversal of a tree.</li> <li>9) Write a program to implement a binary tree and its applications.</li> <li>10) Write a program to implement a graph and its applications.</li> <li>11) Write a program to implement a map and its applications.</li> </ol>

### Semester IV - Theory

<p><b>Course:</b> <b>USCS401</b></p>	<p><b>TOPICS (Credits : 02 Lectures/Week: 03)</b> <b>Operating Systems and Linux</b></p>	
<p><b>Objectives:</b> To provide a sound understanding of Computer operating system, its structures, functioning and algorithms.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) Enable the learner to gain extensive knowledge on principles, functioning and structure of operating systems.</li> <li>2) Appreciate the importance of Operating system as resource manager and gain insight into how computing resources (such as CPU and memory) are managed by the operating system.</li> </ol>		

	<p>3) Learn about process management, process scheduling, threads, synchronization, memory management, virtual memory concepts, cause and effect of deadlocks, and file system</p> <p>4) Learn about Linux system, basic shell command, environmental variables, shell script, structured commands.</p>	
<p><b>Unit I</b></p>	<p><b>Introduction to Operating systems, its structure and Process Scheduling</b> : Definition Operating system, Operating System Services, System Calls and its types, System Programs, Operating system Structure</p> <p><b>Process Management</b>: Concepts, Process Scheduling and Operations, Inter-process Communication, Communication in Client –Server Systems.</p> <p><b>Multithreaded Programming</b> : concepts and benefits, Multithreading Models</p> <p><b>Introduction to LINUX System</b>: The Linux system, Kernel Modules</p> <p><b>bash shell, basic commands</b>: Navigating File system, Listing files and Directories, Handling Files, Managing directories, Viewing File Contents, Monitoring Programs and Disk space, working With data files</p>	<p><b>15L</b></p>
<p><b>Unit II</b></p>	<p><b>Process Scheduling</b> : Scheduling Criteria and algorithms, Thread Scheduling, Multiple processor scheduling, Real-time Scheduling.</p> <p><b>Synchronization, Deadlocks ,Memory and Storage Management</b></p> <p><b>Synchronization</b> : Background,, Critical-Section Problem, Peterson’s Solution, Mutex Locks and Semaphores, Monitors</p> <p><b>Deadlocks</b>: Concept and its characterization, Methods of deadlock handling, Deadlock prevention, Deadlock Avoidance (Safe state and Resource-allocation-graph algorithm), Deadlock Detection, Recovery from Deadlock</p> <p><b>Linux bash commands</b>: Linux environment variables, setting environment variables, Removing environment variables, Variable arrays</p> <p><b>bash scripting</b>: Creating a script file, Displaying messages, Using variables, Redirecting Input and Output, Pipes, Performing math</p>	<p><b>15L</b></p>
<p><b>Unit III</b></p>	<p><b>Memory Management Strategies</b> : Swapping, Contiguous Memory Allocation, Segmentation, Paging and page tables</p> <p><b>Virtual Memory Management</b> : Concept, Demand Paging, Copy-on-Write, Page replacement</p> <p><b>File System</b> : Concept, Access methods, Structure, Directory and disk structure, File system Mounting</p> <p><b>bash structured commands</b>: Working with the if-then, test command, Compound condition testing, Advanced if then features, the case command, for command, until command, while command</p>	<p><b>15L</b></p>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) <i>Operating Systems Concepts</i>- Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, 9th Edition, John Wiley</li> <li>2) <i>Linux Command Line and Shell Scripting BIBLE</i>, Richard Blum, John Wiley</li> <li>3) <i>Official Ubuntu Book 8th Ed Ver 0.2</i>, by Matthew Helmke &amp; Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew</li> </ol>		

**Reference(s):**

- 1) *Operating Systems: Internals and Design Principles*, 8th edition, William Stallings; Prentice Hall.
- 2) *Operating Systems*, 3rd edition, Gary Nutt; Pearson/Addison Wesley.
- 3) *Modern Operating Systems*, 3rd edition, Andrew S. Tanenbaum; Prentice Hall.
- 4) <https://help.ubuntu.com/community/Java>

Course: USCS402	TOPICS (Credits : 02 Lectures/Week: 03) Java Programming	
<p><b>Objective :</b> The objective of this course is to teach the learner how to implement the code in Object oriented environment and understand the concepts of Core java and to cover-up with the pre-requisites of Core java.</p> <p><b>Expected Learning Outcomes :</b> Through this course there will be an enhancement to</p> <ol style="list-style-type: none"> <li>1) Object oriented programming concepts using Java.</li> <li>2) Knowledge of input, its processing and getting suitable output.</li> <li>3) Understand, design, implement and evaluate classes and applets.</li> <li>4) Knowledge and implementation of AWT package.</li> </ol>		
<b>Unit I</b>	<p><b>Introduction to Java and evolution :</b> Classes and Objects, data abstraction and encapsulation, inheritance, polymorphism, dynamic binding and message communication</p> <p><b>Overview of java language :</b> writing simple java program, an application with two classes, structure, tokens, statements, implementing java program, JVM, command line arguments, programming style.</p> <p>Constants, variables and data types, declaration of variable, value and scope of variable, symbolic constants, type casting, getting value of variable and standard default value.</p> <p>Operator and expressions</p> <p><b>Decision making and branching:</b> decision making with if statement, simple if, if...else, nesting of if..... else, else if ladder, Switch statement, ? Operator.</p> <p><b>Decision making and looping :</b> WHILE statement, DO statement, FOR statement, JUMP in loops, labeled loops.</p>	<b>15L</b>
<b>Unit II</b>	<p><b>Working with Classes, objects and methods :</b> Array Strings, Vectors and Wrapper classes. Interfaces, multiple inheritances, Package and its implementation - putting classes together. JAVA API packages.</p> <p><b>Multithreaded programming:</b> Creating Threads, extending the Thread class, stopping and blocking a Thread, life cycle of a Thread, using Thread methods, Thread exceptions, Thread priority, Synchronization,</p>	<b>15L</b>

	<p>implementing “runnable” interface.</p> <p><b>Error and Exception Handling:</b> Managing errors, exceptions and using exception for debugging.</p>	
<b>Unit III</b>	<p><b>Applet programming :</b> Writing Applet, building the code, Applet life cycle, creating an executable Applet and adding Applet to HTML file, running the Applet, passing parameters to an Applet, aligning the display.</p> <p><b>Graphics programming:</b> Introduction to Graphics class, lines, rectangles, circle, ellipses, drawing arcs, polygons, line graphs, using control loops in Applet, drawing bars and charts.</p> <p><b>AWT package:</b> Window fundamentals - Component, Container, Panel, Window, Frame, and Canvas. AWT Controls: Label, Button, TextField, TextArea, CheckBox, CheckBoxGroup, Choice, and List.</p> <p><b>Layout Manager:</b> FlowLayout, BorderLayout, GridLayout.</p> <p><b>Tour of Swing:</b> Introduction to Swing and implementation of its basic components.</p>	<b>15L</b>
<p><b>Text book(s):</b></p> <ol style="list-style-type: none"> <li>1) <i>Programming with java a primer</i> by E-Balagurusamy, Tata Mc graw Hill</li> <li>2) <i>Java 8 Programming Black Book</i>, Dt Editorial Services, Dreamtech Press</li> <li>3) <i>Java The Complete Reference</i> 8th Edition, Herb Schildt, Oracle Press, McGraw Hill Education</li> <li>4) <i>Object Oriented Programming with Java: Essentials and Applications</i>, Tata McGraw Hill</li> </ol> <p><b>Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) <i>Java 8 in Action</i>, Mario Fusco , Raoul - Gabriel Urma , Alan Mycroft</li> <li>2) <i>Java SE 8 for programmers</i>, third edition by paul dietel , Harvey daitel (Deitel Developer Series)</li> </ol>		



<b>Course:</b> <b>USCS403</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b> <b>Web Technologies</b>	
<p><b>Objectives:</b> To provide insight into technologies used to develop web applications and learn about creating, displaying and managing web contents. Explore about Client Side and Server side using Markup and Scripting Languages.</p> <p><b>Expected Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1) To gain knowledge about Markup Languages for developing web applications.</li> <li>2) Enhance Presentation of contents using Style Sheets.</li> <li>3) To understand and implement the Client side validations using Scripting language.</li> <li>4) To understand and implement the Sever side validations using Scripting language.</li> </ol>		
<b>Unit I</b>	<p><b>Web Programming using HTML</b></p> <p><b>Introduction to Web Technologies:</b> HTML Fundamentals, HTML 4.0 elements, and Tags, Attributes, Event Handlers, Document Structure Tags, Working with Text, Formatting Tags, List Tags, links and URLs, Hyperlinks, Image &amp; Image map, color, Table Tags, Form Tags, Frame Tags, Executable Content Tags</p> <p><b>Introduction to HTML 5:</b> Difference between HTML4.0 and HTML5, Features of HTML5, New Tags in HTML5, Working with Multimedia-Use of Audio and Video Tags.</p>	15L
<b>Unit II</b>	<p><b>CSS and Java Script</b></p> <p><b>CSS:</b> Introduction to CSS, CSS Sectors, CSS in HTML, Inline Styles – Embedding Styles- Linking External Style Sheets, Working with background, color, font and text with CSS, Display and positioning an element, Effects, Frames and controls in CSS</p> <p><b>Introduction to Java Script:</b> Features and Fundamentals, Functions, Events, Objects in Java Script, Browser Objects, Java Script in Web Browser, The Document Object Model, Events and Event Handling, Forms and Form Elements.</p>	15L
<b>Unit III</b>	<p><b>Creating and Using PHP Applications</b></p> <p><b>Introduction to PHP:</b> Features and advantages, Creating and running PHP Scripts, Handling errors, Using Variables and Constants, Data Types, Operators, Control Structures, String Functions, Array and Array Functions like \$_GET, \$_POST, \$_REQUEST.</p> <p><b>Controlling Program flow:</b> Conditional and Looping Structures, Break, continue and Exit Statements</p> <p><b>Forms and Database:</b> Web Forms, Working with FORM tag, Form processing and Validations, Working with Databases-PHP and MySQL, connection, Adding, altering, Inserting, Modifying and Retrieving Data</p>	15L

**Textbook(s):**

- 1) *Using HTML 4, XML & JAVA* by Eric Ladd & Jim O'Donnell. (Platinum Edition) (PHI)
- 2) *Web Technologies* -Black Book Series, DT Editorial Service, Dream Tech Press.
- 3) *HTML5 Black Book: Covers CSS3, JAVASCRIPT, XML,XHTML, AJAX, PHP and JQUERY* DreamTech Press.
- 4) *Beginning PHP6, Apache, MySQL Web Development*, Timothy Boronczyk, Wrox Publication.

**Reference(s):**

- 1) *Murach's HTML5 and CSS3* by Zak Ruvalcaba, Anne Bohem, Shroff Publishers and Distributors
- 2) *Web Technology*, Ralph Moseley, Wiley India
- 3) *HTML 5 for Beginners*, Firuza Aibra, Shroff Publishers and Distributors
- 4) Mike Mcgrath, "*PHP & MySQL in Easy Steps*", Tata McGraw Hill
- 5) *Learning PHP, MySQL, JavaScript, CSS & HTML5*, 3rd Edition by Robin Nixon

## Semester IV Practical

USCSP04	<b>Practical/Tutorial of USCS301 + USCS302 + USCS303 (Credits: 03, Pract/Week: 09)</b>
	<p><b>Operating System and Linux :</b></p> <ol style="list-style-type: none"> <li>1) <b>Installation of Ubuntu Linux operating system.</b> <ol style="list-style-type: none"> <li>a) Booting and Installing from (USB/DVD)</li> <li>b) Installing from the Minimal CD</li> </ol> </li> <li>2) <b>Introduction to bash:</b> Basic shell commands for directory and file manipulation, like ls, cd, pwd, cp, mv, rm</li> <li>3) <b>Finding and Installing Ubuntu Applications</b> <ol style="list-style-type: none"> <li>a) Using Ubuntu software center</li> <li>b) Using Synaptic</li> <li>c) Explore useful software packages.</li> </ol> </li> <li>4) <b>More bash commands:</b> like echo, history, date, chmod, who, man</li> <li>5) <b>Customizing Ubuntu for performance , Accessibility and Fun</b> <ol style="list-style-type: none"> <li>a) Appearance Tool</li> <li>b) Unity Tweak Tool</li> <li>c) Compiz Config Setting Manager</li> <li>d) Unity Lenses and scope</li> </ol> </li> <li>6) <b>Shell scripting I</b> <ol style="list-style-type: none"> <li>a) defining variables, reading user input</li> <li>b) Conditions (if - then, case) arithmetic operations</li> </ol> </li> <li>7) <b>Becoming an Ubuntu power user</b> <ol style="list-style-type: none"> <li>a) Administering system and User setting</li> <li>b) Learning Unity keyboard Shortcuts</li> <li>c) Using the Terminal</li> <li>d) Working with windows programs</li> </ol> </li> <li>8) Working with data files (sort, grep, linux File compression Utilities –bzip2,</li> </ol>

- gzip, zip, Archiving data- tar)
- 9) **Shell scripting II**  
Conditions (for loop, until loop and while loop) arithmetic operations
  - 10) **Shell scripting III-** Redirecting Output in Scripts , Redirecting Input in Scripts, Creating Your Own Redirection
  - 11) Working and managing with processes: sh, ps, kill, nice, at and batch etc.
  - 12) **Using javac compiler**

**Java Programming :**

- 1) Write java programs to demonstrate following
  - a) A single class
  - b) Multiple classes
  - c) Use of command line arguments
  - d) Read data from keyboard.
  - e) Creating and casting of variables.
- 2) Write java programs to implement various operators and functions using arithmetic expression
  - a) Arithmetic operators
  - b) Relational operator
  - c) Logical operator
  - d) Assignment operator
  - e) Increment and decrement operator
  - f) Conditional operator
  - g) Bitwise and special operator
  - h) Arithmetic operators
  - i) Mathematical functions.
- 3) Write Java program that illustrates the concepts of selection statement, looping, nested loops, breaking out of loop.
- 4) Write Java programs that illustrates the concepts of Java class that includes
  - a) Constructor with and without parameters, destructor
  - b) Overloading methods.
  - c) Creating objects
  - d) Static members
- 5) Write Java program to demonstrate inheritance by creating suitable classes.
- 6) Write Java program that illustrates the concepts of one and two dimension arrays and strings.
- 7) Write Java programs that illustrates the concept of following:-
  - a) Interface in java
  - b) Package in java
  - c) Multithreading in Java
- 8) Write program that illustrates the error handling using exception handling.
- 9) Write Java applet to demonstrate graphics, Font and Color classes.
- 10) Write Java program to illustrate AWT package, Event, classes and listeners and swing package.

**Web Technologies :**

**Write a program to**

- 1) Create a web applications-Frame Tags and Image Mapping
- 2) Create a web applications-List tags, Table tags
- 3) Create and use a web application-Form Tags
- 4) Create application using HTML5 Tags -Audio Tags and Video Tags
- 5) Apply CSS(Internal and External style) to a Web Page
- 6) Execute different Control structures using JavaScript
- 7) Execute Client Side Scripts using JavaScript
- 8) Retrieve data from HTML form using PHP
- 9) Retrieve Employee Details/ Registration Details from the database using PHP
- 10) Add, Modify and Delet data from client side into table in MySQL

**Note:** Use of Freeware\Open source software for server is recommended

# Evaluation Scheme

## I. Internal Exam - 25 Marks

### (i) Test – 20 Marks

20 marks Test – Duration 40 mins

It will be conducted either using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment) Or a test based on an equivalent online course on the contents of the concerned course(subject) offered by or build using MOOC (Massive Open Online Course) platform.

- (ii) 5 Marks – Active participation in routine class instructional deliveries  
Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

## II. External Exam - 75 Marks

1. Duration - 2.5 Hours.
2. Theory question paper pattern:-

All questions are compulsory.			
Question	Based on		Marks
Q.1	Unit I		[20]
Q.2	Unit II		[20]
Q.3	Unit III		[20]
Q.4	Unit I, II and III		[15]

All questions shall be compulsory with internal choice within the questions.

Each Question may be sub divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

## III. Practical Exam –

- Each course carry 50 Marks : 40 marks + 05 marks (journal) + 05 marks (viva)
- Minimum 75 % practical from each paper are required to be completed and written in the journal.

**(Certified Journal is compulsory for appearing at the time of Practical Exam)**

(2)

**UNIVERSITY OF MUMBAI**

No. UG/107bf 2017

**CIRCULAR:-**

A reference is invited to the Syllabi relating to the B.Sc. degree course, **vide** this office Circular No. UG/42 of 2016-17, dated 5<sup>th</sup> August, 2016 and the Principals of the affiliated Colleges in Science are hereby informed that the recommendation made by Ad-hoc-Board of Studies in Computer Science at its meeting held on 5/5/2017 has been accepted by the Academic Council at its meeting held on 11.5.2017 **vide** item No. 4.210 and that in accordance therewith, in revised syllabus as per the Credit Based Semester and Grading System for S.Y.B.Sc Computer Science (Sem III & IV) which is available on the University's website ([www.mu.ac.in](http://www.mu.ac.in)) and that the same has been brought into force with effect from the academic year 2016-17.

MUMBAI – 400 032

२५/७/१७ July, 2017

*Alam*  
25/7/17  
REGISTRAR

To,

The Principal of the affiliated Colleges in Science and the Head of Recognized Institutions concerned.

**A.C/4.210/11.05.2017**

No. UG/107-A of 2017

MUMBAI-400 032

२५/७/१७ July, 2017

Copy forwarded with compliments for information to :-

- 1) The Co-ordinator, Faculty of Science,
- 2) The Offg. Director of Board of Examinations and Evaluation,
- 3) The Chairperson, Board of Studies in Botany,
- 4) The Director of Board of Studies Development,
- 5) The Professor-cum-Director, Institute of Distance and Open Learning.
- 6) The Co-Ordinator, University Computerization Centre.

*Alam*  
25/7/17  
REGISTRAR

....PTO

AC\_\_\_\_\_

**UNIVERSITY OF MUMBAI**



**Syllabus for Sem III & IV**  
**Program: B.Sc.**  
**Course: Computer Science**

(Credit Based Semester and Grading System with  
effect from the academic year 2017-2018 )

## Preamble

The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skills sets demanded under new technological environment. It also endeavours to align the programme structure and course curriculum with student aspirations and corporate expectations. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present day context.

Second year of this course is about studying core computer science subjects. Theory of Computation course provides understanding of grammar, syntax and other elements of modern language designs. It also covers developing capabilities to design formulations of computing models and its applications in diverse areas.

The course in Operating System satisfies the need of understanding the structure and functioning of system. Programming holds key indispensable position in any curriculum of Computer Science. It is essential for the learners to know how to use object oriented paradigms. There is also one dedicated course Android Developer Fundamentals as a skill enhancement catering to modern day needs of Mobile platforms and applications. The syllabus has Database Systems courses in previous semesters. The course in Database Management Systems is its continuation in third semester. The course has objectives to develop understanding of concepts and techniques for data management along with covers concepts of database at advance level.

The course of Combinatorics and Graph Theory in third semester and the course of Linear Algebra in fourth semester take the previous courses in Mathematics. Graph theory is rapidly moving into the mainstream mainly because of its applications in diverse fields which include can further open new opportunities in the areas of genomics, communications networks and coding theory, algorithms and computations and operations research.

Introducing one of the upcoming concepts Physical Computing and IoT programming will definitely open future area as Embedded Engineer, involvement in IoT projects, Robotics and many more. The RasPi is a popular platform as it offers a complete Linux server in a tiny platform for a very low cost and custom-built hardware with minimum complex hardware builds which is easier for projects in education domain.



**S.Y.B.Sc. (Semester III and IV)**  
**Computer Science Syllabus**  
**Credit Based Semester and Grading System**  
**To be implemented from the Academic year 2017-2018**

<b>SEMESTER III</b>			
<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>L / Week</b>
USCS301	Theory of Computation	2	3
USCS302	Core JAVA	2	3
USCS303	Operating System	2	3
USCS304	Database Management Systems	2	3
USCS305	Combinatorics and Graph Theory	2	3
USCS306	Physical Computing and IoT Programming	2	3
USCS307	Skill Enhancement: Web Programming	2	3
USCSP301	USCS302+USCS303+USCS304	3	9
USCSP302	USCS305+USCS306+USCS307	3	9

<b>SEMESTER IV</b>			
<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>L / Week</b>
USCS401	Fundamentals of Algorithms	2	3
USCS402	Advanced JAVA	2	3
USCS403	Computer Networks	2	3
USCS404	Software Engineering	2	3
USCS405	Linear Algebra using Python	2	3
USCS406	.NET Technologies	2	3
USCS407	Skill Enhancement: Android Developer Fundamentals	2	3
USCSP401	USCS401+ USCS402+ USCS403	3	9
USCSP402	USCS405+ USCS406+ USCS407	3	9

## SEMESTER III

### THEORY

<b>Course:</b> <b>USCS301</b>	<b>TOPICS (Credits : 02 Lectures/Week:03)</b> <b>Theory of Computation</b>	
<b>Objectives:</b> To provide the comprehensive insight into theory of computation by understanding grammar, languages and other elements of modern language design. Also to develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas.		
<b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"><li>1. Understand Grammar and Languages</li><li>2. Learn about Automata theory and its application in Language Design</li><li>3. Learn about Turing Machines and Pushdown Automata</li><li>4. Understand Linear Bound Automata and its applications</li></ol>		
<b>Unit I</b>	<b>Automata Theory:</b> Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NFA equivalence, Mealy and Moore Machines, Minimizing Automata. <b>Formal Languages:</b> Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata	<b>15L</b>
<b>Unit II</b>	<b>Regular Sets and Regular Grammar:</b> Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar <b>Context Free Languages:</b> Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG <b>Pushdown Automata:</b> Definitions, Acceptance by PDA, PDA and CFG	<b>15L</b>

<b>Unit III</b>	<p><b>Linear Bound Automata:</b> The Linear Bound Automata Model, Linear Bound Automata and Languages.</p> <p><b>Turing Machines:</b> Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine,</p> <p><b>Undecidability:</b> The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems</p>	<b>15L</b>
<p><b>Tutorials :</b></p> <ol style="list-style-type: none"> <li>1. Problems on generating languages for given simple grammar</li> <li>2. Problems on DFA and N DFA equivalence</li> <li>3. Problems on generating Regular Expressions</li> <li>4. Problems on drawing transition state diagrams for Regular Expressions</li> <li>5. Problems on Regular Sets and Regular Grammar</li> <li>6. Problems on Ambiguity of Grammar</li> <li>7. Problems on working with PDA</li> <li>8. Problems on working with Turing Machines</li> <li>9. Problems on generating derivation trees</li> <li>10. Problems on Linear Bound Automata/Universal Turing Machine</li> </ol>		
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI, 3<sup>rd</sup> Edition</li> <li>2) Introduction to Computer Theory, Daniel Cohen, Wiley, 2<sup>nd</sup> Edition</li> <li>3) Introductory Theory of Computer Science, E.V. Krishnamurthy, Affiliated East-West Press.</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Theory of Computation, Kavi Mahesh, Wiley India</li> <li>2) Elements of The Theory of Computation, Lewis, Papadimitriou, PHI</li> <li>3) Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education</li> <li>4) Introduction to Theory of Computation, Michel Sipser, Thomson</li> </ol>		

<b>Course:</b>  <b>USCS302</b>	<b>TOPICS (Credits : 02 Lectures/Week:03)</b>  <b>Core Java</b>	
<b>Objectives:</b> <p>The objective of this course is to teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and to cover-up with the pre-requisites of Core java.</p> <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. Object oriented programming concepts using Java.</li> <li>2. Knowledge of input, its processing and getting suitable output.</li> <li>3. Understand, design, implement and evaluate classes and applets.</li> <li>4. Knowledge and implementation of AWT package.</li> </ol>		
<b>Unit I</b>	<b>The Java Language:</b> Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays <b>OOPS:</b> Introduction, Class, Object, Static Keywords, Constructors, this Key Word, Inheritance, super Key Word, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces <b>String Manipulations:</b> String, String Buffer, String Tokenizer <b>Packages:</b> Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers	<b>15L</b>
<b>Unit II</b>	<b>Exception Handling:</b> Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples <b>Multithreading:</b> Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods <b>I/O Streams:</b> Introduction, Byte-oriented streams, Character- oriented streams, File, Random access File, Serialization <b>Networking:</b> Introduction, Socket, Server socket, Client –Server Communication	<b>15L</b>
	<b>Wrapper Classes:</b> Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes <b>Collection Framework:</b> Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes	

<b>Unit III</b>	<p><b>Inner Classes:</b> Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class</p> <p><b>AWT:</b> Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, CheckBox, Radio Button, Choice, List, Menu, Text Field, Text Area</p>	<b>15L</b>
-----------------	---	------------

**Textbook(s):**

- 1) Herbert Schildt, Java The Complete Reference, Ninth Edition, McGraw-Hill Education, 2014

**Additional Reference(s):**

- 1) E. Balagurusamy, Programming with Java, Tata McGraw-Hill Education India, 2014
- 2) Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press
- 3) The Java Tutorials: <http://docs.oracle.com/javase/tutorial/>

<b>Course:</b> <b>USCS303</b>	<b>TOPICS (Credits : 02 Lectures/Week:03)</b> <b>Operating System</b>	
<b>Objectives:</b>		
Learners must understand proper working of operating system. To provide a sound understanding of Computer operating system, its structures, functioning and algorithms.		
<b>Expected Learning Outcomes:</b>		
<ol style="list-style-type: none"> <li>1. To provide a understanding of operating system, its structures and functioning</li> <li>2. Develop and master understanding of algorithms used by operating systems for various purposes.</li> </ol>		
<b>Unit I</b>	<p><b>Introduction and Operating-Systems Structures:</b> Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments</p> <p><b>Operating-System Structures:</b> Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure</p> <p><b>Processes:</b> Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication</p>	<b>15L</b>

	<b>Threads:</b> Overview, Multicore Programming, Multithreading Models	
<b>Unit II</b>	<p><b>Process Synchronization:</b> General structure of a typical process, race condition, The Critical-Section Problem, Peterson’s Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors</p> <p><b>CPU Scheduling:</b> Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling</p> <p><b>Deadlocks:</b> System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Main Memory:</b> Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table</p> <p><b>Virtual Memory:</b> Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing</p> <p><b>Mass-Storage Structure:</b> Overview, Disk Structure, Disk Scheduling, Disk Management</p> <p><b>File-System Interface:</b> File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing</p> <p><b>File-System Implementation:</b> File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 8<sup>th</sup> Edition</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill</li> <li>2. Naresh Chauhan, Principles of Operating Systems, Oxford Press</li> <li>3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016</li> </ol>		

<b>Course:</b> <b>USCS304</b>	<b>TOPICS (Credits : 02 Lectures/Week:03)</b> <b>Database Management Systems</b>	
<b>Objectives:</b> To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage. <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. Master concepts of stored procedure and triggers and its use.</li> <li>2. Learn about using PL/SQL for data management</li> <li>3. Understand concepts and implementations of transaction management and crash recovery</li> </ol>		
<b>Unit I</b>	<p><b>Stored Procedures:</b> Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures.</p> <p><b>Triggers:</b> Concept of triggers, Implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers.</p> <p><b>Sequences:</b> creating sequences, referencing, altering and dropping a sequence.</p> <p><b>File Organization and Indexing:</b> Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes.</p>	<b>15L</b>
	<p><b>Fundamentals of PL/SQL:</b> Defining variables and constants, PL/SQL expressions and comparisons: Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements, PL/SQL Datatypes: Number Types, Character Types, Boolean Type, Datetime and Interval Types.</p>	

<b>Unit II</b>	<b>Overview of PL/SQL Control Structures:</b> Conditional Control: IF and CASE Statements, IF-THEN Statement, IF-THEN-ELSE Statement, IFTHEN-ELSIF Statement, CASE Statement, Iterative Control: LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP, Sequential Control: GOTO and NULL Statements	<b>15L</b>
<b>Unit III</b>	<p><b>Transaction Management:</b> ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem, Inconsistent Read Problem , Read-Write Locks, Deadlocks Handling, Two Phase Locking protocol.</p> <p><b>DCL Statements:</b> Defining a transaction, Making Changes Permanent with COMMIT, Undoing Changes with ROLLBACK, Undoing Partial Changes with SAVEPOINT and ROLLBACK</p> <p><b>Crash Recovery:</b> ARIES algorithm. The log based recovery, recovery related structures like transaction and dirty page table, Write-ahead log protocol, check points, recovery from a system crash, Redo and Undo phases.</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Ramakrishnam, Gehrke, Database Management Systems, Bayross, McGraw-Hill,3<sup>rd</sup> Edition</li> <li>2) Abraham Silberschatz, Henry F. Korth,S. Sudarshan , Database System Concepts, 6<sup>th</sup> Edition</li> <li>3) Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Ramez Elmasri &amp; Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education</li> <li>2) Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press.</li> <li>3) Joel Murach, Murach’s MySQL, Murach</li> </ol>		



<b>Course:</b> <b>USCS305</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b> <b>Combinatorics and Graph Theory</b>	
<b>Objectives:</b> To give the learner a broad exposure of combinatorial Mathematics through applications especially the Computer Science applications. <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. Appreciate beauty of combinatorics and how combinatorial problems naturally arise in many settings.</li> <li>2. Understand the combinatorial features in real world situations and Computer Science applications.</li> <li>3. Apply combinatorial and graph theoretical concepts to understand Computer Science concepts and apply them to solve problems</li> </ol>		
<b>Unit I</b>	<b>Introduction to Combinatorics:</b> Enumeration, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, Sudoku Puzzles. Strings, Sets, and Binomial Coefficients: Strings- A First Look, Combinations, Combinatorial, The Ubiquitous Nature of Binomial Coefficients, The Binomial, Multinomial Coefficients. Induction: Introduction, The Positive Integers are Well Ordered, The Meaning of Statements, Binomial Coefficients Revisited, Solving Combinatorial Problems Recursively, Mathematical Induction, and Inductive Definitions Proofs by Induction. Strong Induction	<b>15L</b>
<b>Unit II</b>	<b>Graph Theory:</b> Basic Notation and Terminology, Multigraphs: Loops and Multiple Edges, Eulerian and Hamiltonian Graphs, Graph Coloring, Planar Counting, Labeled Trees, A Digression into Complexity Theory. Applying Probability to Combinatorics, Small Ramsey Numbers, Estimating Ramsey Numbers, Applying Probability to Ramsey Theory, Ramsey's Theorem The Probabilistic Method	<b>15L</b>
<b>Unit III</b>	<b>Network Flows:</b> Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm,	<b>15L</b>

	A Concrete Example, Integer Solutions of Linear Programming Problems. Combinatorial Applications of Network Flows: Introduction, Matching in Bipartite Graphs, Chain partitioning, Pólya's Enumeration Theorem: Coloring the Vertices of a Square.	
--	--	--

**Textbook(s):**

- 1) Applied Combinatorics, Mitchel T. Keller and William T. Trotter, 2016, <http://www.rellek.net/appcomb>.

**Additional Reference(s):**

- 1) Applied Combinatorics, sixth.edition, Alan Tucker, Wiley; (2016)
- 2) Graph Theory and Combinatorics, Ralph P. Grimaldi, Pearson Education; Fifth edition (2012)
- 3) Combinatorics and Graph Theory, John Harris, Jeffrey L. Hirst, Springer( 2010).
- 4) Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008).

<b>Course:</b> <b>USCS306</b>	<b>TOPICS (Credits : 02 Lectures/Week:03)</b> <b>Physical Computing and IoT Programming</b>
----------------------------------	--

**Objectives:**

To learn about SoC architectures; Learn how Raspberry Pi. Learn to program Raspberry Pi. Implementation of internet of Things and Protocols.

**Expected Learning Outcomes:**

1. Enable learners to understand System On Chip Architectures.
2. Introduction and preparing Raspberry Pi with hardware and installation.
3. Learn physical interfaces and electronics of Raspberry Pi and program them using practical's
4. Learn how to make consumer grade IoT safe and secure with proper use of protocols.

<p><b>Unit I</b></p>	<p><b>SoC and Raspberry Pi</b>  <b>System on Chip:</b> What is System on chip? Structure of System on Chip.  <b>SoC products:</b> FPGA, GPU, APU, Compute Units.  <b>ARM 8 Architecture:</b> SoC on ARM 8. ARM 8 Architecture Introduction  <b>Introduction to Raspberry Pi:</b> Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi.  <b>Raspberry Pi Boot:</b> Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.</p>	<p><b>15L</b></p>
<p><b>Unit II</b></p>	<p><b>Programming Raspberry Pi</b>  <b>Raspberry Pi and Linux:</b> About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands  <b>Programing interfaces:</b> Introduction to Node.js, Python.  <b>Raspberry Pi Interfaces:</b> UART, GPIO, I2C, SPI  <b>Useful Implementations:</b> Cross Compilation, Pulse Width Modulation, SPI for Camera.</p>	<p><b>15L</b></p>
<p><b>Unit III</b></p>	<p><b>Introduction to IoT:</b> What is IoT? IoT examples, Simple IoT LED Program.  <b>IoT and Protocols</b>  <b>IoT Security:</b> HTTP, UPnp, CoAP, MQTT, XMPP.  <b>IoT Service as a Platform:</b> Clayster, Thinger.io, SenseIoT, carriots and Node RED.  <b>IoT Security and Interoperability:</b> Risks, Modes of Attacks, Tools for Security and Interoperability.</p>	<p><b>15L</b></p>
<p><b>Textbook(s):</b>  1) Learning Internet of Things, Peter Waher, Packt Publishing(2015)  2) Mastering the Raspberry Pi, Warren Gay, Apress(2014)</p> <p><b>Additional Reference(s):</b>  1) Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly</p>		

<b>Course:</b> <b>USCS307</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b>  <b>Web Programming</b>	
<b>Objectives:</b> <p>To provide insight into emerging technologies to design and develop state of - the art web applications using client-side scripting, server-side scripting, and database connectivity.</p> <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. To design valid, well-formed, scalable, and meaningful pages using emerging technologies.</li> <li>2. Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites</li> <li>3. To develop and implement client-side and server-side scripting language programs.</li> <li>4. To develop and implement Database Driven Websites.</li> <li>5. Design and apply XML to create a markup language for data and document centric applications.</li> </ol>		
<b>Unit I</b>	<b>HTML5:</b> Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page  <b>CSS:</b> Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element	<b>15L</b>
<b>Unit II</b>	<b>JavaScript:</b> Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript – Variables, Operators, Control Flow Statements, Popup Boxes, Functions – Defining and Invoking a Function, Defining Function arguments, Defining a Return Statement, Calling Functions with Timer, JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window, Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript  <b>XML:</b> Comparing XML with HTML, Advantages and Disadvantages of XML,	<b>15L</b>

	Structure of an XML Document, XML Entity References, DTD, XSLT: XSLT Elements and Attributes - xsl:template, xsl:apply-templates, xsl:import, xsl:call-template, xsl:include, xsl:element, xsl:attribute, e xsl:attribute-set, xsl:value-of	
<b>Unit III</b>	<p><b>AJAX:</b> AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties and Methods, Handling asynchronous requests using AJAX</p> <p><b>PHP:</b> Variables and Operators, Program Flow, Arrays, Working with Files and Directories, Working with Databases, Working with Cookies, Sessions and Headers</p> <p><b>Introduction to jQuery:</b> Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects</p>	<b>15L</b>
<p><b>Text Book(s):</b></p> <ol style="list-style-type: none"> <li>1) HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press</li> <li>2) Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India.</li> <li>3) PHP: A Beginners Guide, Vikram Vaswani, TMH</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY</li> <li>2) Learn to Master HTML 5, scriptDemics, StarEdu Solutions Pvt Ltd.</li> <li>3) Learning PHP, MySQL, JavaScript, CSS &amp; HTML5, Robin Nixon, O’Reilly</li> <li>4) PHP, MySQL, JavaScript &amp; HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley</li> </ol>		

## Suggested List of Practical- SEMESTER III

<b>Course:</b> <b>USCSP301</b>	<b>(Credits : 03 Lectures/Week: 09)</b> <b>USCS302+ USCS303+USCS304</b>	
<b>USCS302: Core JAVA</b>		
<ol style="list-style-type: none"> <li>1. Accept integer values for a, b and c which are coefficients of quadratic equation. Find the solution of quadratic equation.</li> <li>2. Accept two n x m matrices. Write a Java program to find addition of these matrices.</li> <li>3. Accept n strings. Sort names in ascending order.</li> <li>4. Create a package: Animals. In package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package animals.</li> <li>5. Demonstrate Java inheritance using extends keyword.</li> <li>6. Demonstrate method overloading and method overriding in Java.</li> <li>7. Demonstrate creating your own exception in Java.</li> <li>8. Using various swing components design Java application to accept a student's resume. (Design form)</li> <li>9. Write a Java List example and demonstrate methods of Java List interface.</li> <li>10. Design simple calculator GUI application using AWT components.</li> </ol>		
<b>USCS303: Operating System</b>		
<i>Practical can be implemented either in JAVA or any other programming language.</i>		
<ol style="list-style-type: none"> <li>1. <b>Process Communication:</b> <ol style="list-style-type: none"> <li>(i) Give solution to the producer–consumer problem using shared memory.</li> <li>(ii) Give solution to the producer–consumer problem using message passing.</li> <li>(iii) One form of communication in a Client–Server Systems environment is Remote method invocation (RMI). RMI is a Java feature similar to RPCs. RMI allows a thread to invoke a method on a remote object. Objects are considered remote if they reside in a different Java virtual machine (JVM). Demonstrate RMI program for adding/subtracting/multiplying/dividing two numbers.</li> </ol> </li> <li>2. <b>Threads:</b> <ol style="list-style-type: none"> <li>(i) The Java version of a multithreaded program that determines the summation of a</li> </ol> </li> </ol>		

non-negative integer. The Summation class implements the Runnable interface. Thread creation is performed by creating an object instance of the Thread class and passing the constructor a Runnable object.

- (ii) Write a multithreaded Java program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.
- (iii) The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, ... Formally, it can be expressed as:  $fib_0 = 0$ ,  $fib_1 = 1$ ,  $fib_n = fib_{n-1} + fib_{n-2}$  Write a multithreaded program that generates the Fibonacci sequence using either the Java,

### 3. Synchronization:

- (i) Give Java solution to Bounded buffer problem.
- (ii) Give solution to the readers–writers problem using Java synchronization.
- (iii) The Sleeping-Barber Problem: A barber shop consists of awaiting room with  $n$  chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers using Java synchronization.

4. Implement FCFS scheduling algorithm in Java.

5. Implement SJF (with no preemption) scheduling algorithm in Java

6. Implement RR scheduling algorithm in Java

7. Write a Java program that implements the banker's algorithm

8. Write a Java program that implements the FIFO page-replacement algorithm.

9. Write a Java program that implements the LRU page-replacement algorithm.

10. Design a File System in Java.

## USCS304: Database Management Systems

1. Creating and working with Insert/Update/Delete Trigger using Before/After clause.
2. Writing PL/SQL Blocks with basic programming constructs by including following:
  - a. Sequential Statements
  - b. unconstrained loop
3. Sequences:
  - a. Creating simple Sequences with clauses like START WITH, INCREMENT BY, MAXVALUE, MINVALUE, CYCLE | NOCYCLE, CACHE | NOCACHE, ORDER | NOORDER.
  - b. Creating and using Sequences for tables.
4. Writing PL/SQL Blocks with basic programming constructs by including following:
  - a. If...then...Else, IF...ELSIF...ELSE... END IF
  - b. Case statement
5. Writing PL/SQL Blocks with basic programming constructs for following Iterative Structure:
  - a. While-loop Statements
  - b. For-loop Statements.
6. Writing PL/SQL Blocks with basic programming constructs by including a GoTO to jump out of a loop and NULL as a statement inside IF
7. Writing Procedures in PL/SQL Block
  - a. Create an empty procedure, replace a procedure and call procedure
  - b. Create a stored procedure and call it
  - c. Define procedure to insert data
  - d. A forward declaration of procedure
8. Writing Functions in PL/SQL Block.
  - a. Define and call a function
  - b. Define and use function in select clause,
  - c. Call function in dbms\_output.put\_line
  - d. Recursive function
  - e. Count Employee from a function and return value back
  - f. Call function and store the return value to a variable
9. Writing a recursive Functions in PL/SQL Block
10. Study of transactions and locks



<b>Course:</b> <b>USCSP302</b>	<b>(Credits : 03 Lectures/Week: 09)</b> <b>USCS305+ USCS306+USCS307</b>	
<b>USCS305: Combinatorics and Graph Theory</b>		
<ol style="list-style-type: none"> <li>1. Solving problems on strings, sets and binomial coefficients.</li> <li>2. Solving problems using induction.</li> <li>3. Solving problems on Eulerian and Hamiltonian graphs.</li> <li>4. Solving problems on Chromatic number and coloring</li> <li>5. Solving problems using Kruskal’s Algorithm</li> <li>6. Solving problems using Prim’s Algorithm</li> <li>7. Solving problems using Dijkstra’s Algorithm</li> <li>8. Solving problems of finding augmenting paths in network flows.</li> <li>9. Solving problems on network flows using Ford-Fulkerson Labeling Algorithm</li> <li>10. Solving problems on posets and their associated networks.</li> </ol>		
<b>USCS306: Physical Computing and IoT Programming</b>		
<ol style="list-style-type: none"> <li>1. Preparing Raspberry Pi: Hardware preparation and Installation</li> <li>2. Linux Commands: Exploring the Raspbian</li> <li>3. GPIO: Light the LED with Python</li> <li>4. GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas</li> <li>5. SPI: Camera Connection and capturing Images using SPI</li> <li>6. Real Time Clock display using PWM.</li> <li>7. Stepper Motor Control: PWM to manage stepper motor speed.</li> <li>8. Node RED: Connect LED to Internet of Things</li> <li>9. Stack of Raspberry Pi for better Computing and analysis</li> <li>10. Create a simple Web server using Raspberry Pi</li> </ol>		
<b>USCS307: Web Programming</b>		
<ol style="list-style-type: none"> <li>1. Design a webpage that makes use of <ol style="list-style-type: none"> <li>a. Document Structure Tags</li> <li>b. Various Text Formatting Tags</li> <li>c. List Tags</li> <li>d. Image and Image Maps</li> </ol> </li> <li>2. Design a webpage that makes use of <ol style="list-style-type: none"> <li>a. Table tags</li> <li>b. Form Tags (forms with various form elements)</li> </ol> </li> </ol>		

- c. Navigation across multiple pages
  - d. Embedded Multimedia elements
3. Design a webpage that make use of Cascading Style Sheets with
  - a. CSS properties to change the background of a Page
  - b. CSS properties to change Fonts and Text Styles
  - c. CSS properties for positioning an element
4. Write JavaScript code for
  - a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
  - b. Validating the various Form Elements
5. Write JavaScript code for
  - a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date
  - b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document,
  - c. Storing and Retrieving Cookies
6. Create a XML file with Internal / External DTD and display it using
  - a. CSS
  - b. XSL
7. Design a webpage to handle asynchronous requests using AJAX on
  - a. Mouseover
  - b. button click
8. Write PHP scripts for
  - a. Retrieving data from HTML forms
  - b. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
  - c. Working with Arrays
  - d. Working with Files (Reading / Writing)
9. Write PHP scripts for
  - a. Working with Databases (Storing Records / Retrieving Records and Display them)
  - b. Storing and Retrieving Cookies
  - c. Storing and Retrieving Sessions
10. Design a webpage with some jQuery animation effects.

## SEMESTER IV

### THEORY

<b>Course:</b> <b>USCS401</b>	<b>TOPICS (Credits : 02 Lectures/Week:03)</b> <b>Fundamentals of Algorithms</b>	
<b>Objectives:</b> <ol style="list-style-type: none"><li>1. To understand basic principles of algorithm design and why algorithm analysis is important</li><li>2. To understand how to implement algorithms in Python</li><li>3. To understand how to transform new problems into algorithmic problems with efficient solutions</li><li>4. To understand algorithm design techniques for solving different problems</li></ol> <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"><li>1. Understand the concepts of algorithms for designing good program</li><li>2. Implement algorithms using Python</li></ol>		
<b>Unit I</b>	Introduction to algorithm, Why to analysis algorithm, Running time analysis, How to Compare Algorithms, Rate of Growth, Commonly Used Rates of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega- $\Omega$ Notation, Theta- $\Theta$ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Performance characteristics of algorithms, Master Theorem for Divide and Conquer, Divide and Conquer Master Theorem: Problems & Solutions, Master Theorem for Subtract and Conquer Recurrences, Method of Guessing and Confirming	<b>15L</b>
<b>Unit II</b>	Tree algorithms: What is a Tree? Glossary, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals, Expression Trees, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees Graph Algorithms: Introduction, Glossary, Applications of Graphs, Graph Representation, Graph Traversals, Topological Sort, Shortest Path Algorithms, Minimal Spanning Tree	<b>15L</b>

	Selection Algorithms: What are Selection Algorithms? Selection by Sorting, Partition-based Selection Algorithm, Linear Selection Algorithm - Median of Medians Algorithm, Finding the K Smallest Elements in Sorted Order	
<b>Unit III</b>	<p>Algorithms Design Techniques: Introduction, Classification, Classification by Implementation Method, Classification by Design Method</p> <p>Greedy Algorithms: Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and Disadvantages of Greedy Method, Greedy Applications, Understanding Greedy Technique</p> <p>Divide and Conquer Algorithms: Introduction, What is Divide and Conquer Strategy? Divide and Conquer Visualization, Understanding Divide and Conquer, Advantages of Divide and Conquer, Disadvantages of Divide and Conquer, Master Theorem, Divide and Conquer Applications</p> <p>Dynamic Programming: Introduction, What is Dynamic Programming Strategy? Properties of Dynamic Programming Strategy, Problems which can be solved using Dynamic Programming, Dynamic Programming Approaches, Examples of Dynamic Programming Algorithms, Understanding Dynamic Programming, Longest Common Subsequence</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi , CareerMonk Publications, 2016</li> <li>2. Introduction to Algorithm, Thomas H Cormen, PHI</li> </ol> <p><b>Additional References(s):</b></p> <ol style="list-style-type: none"> <li>1. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 2016, Wiley</li> <li>2. Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran Ellis Horowitz, Universities Press</li> </ol>		

<b>Course:</b> <b>USCS402</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b>  <b>Advanced Java</b>	
<b>Objectives:</b> Explore advanced topic of Java programming for solving problems.  <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1) Understand the concepts related to Java Technology</li> <li>2) Explore and understand use of Java Server Programming</li> </ol>		
<b>Unit I</b>	<b>Swing:</b> Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: JLabel, JTextField and JPasswordField, JTextAres, JButton, JCheckBox, JRadioButton, JComboBox and JList  <b>JDBC:</b> Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updatations, CallableStatement, BLOB & CLOB	<b>15L</b>
<b>Unit II</b>	<b>Servlets:</b> Introduction, Web application Architecture, Http Protocol & Http Methods, Web Server & Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, Servlet Communication, Session Tracking Mechanisms  <b>JSP:</b> Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions,	<b>15L</b>
<b>Unit III</b>	<b>Java Beans:</b> Introduction, JavaBeans Properties, Examples  <b>Struts 2:</b> Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL  <b>JSON:</b> Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java	<b>15L</b>

**Textbook(s):**

- 1) Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR,9<sup>th</sup> Edition
- 2) Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5<sup>th</sup> Edition
- 3) Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3<sup>rd</sup> Edition

**Additional Reference(s):**

- 1) Advanced Java Programming, Uttam K. Roy, Oxford University Press
- 2) *The Java Tutorials: <http://docs.oracle.com/javase/tutorial/>*
- 3) The Java Tutorials of Sun Microsystems Inc

<b>Course:</b> <b>USCS403</b>	<b>TOPICS (Credits :02 Lectures/Week:03)</b> <b>Computer Networks</b>	
<p><b>Objectives:</b></p> <p>In this era of Information, its computation and its exchange techniques, Learner should be able to conceptualize and understand the framework and working of communication networks. And on completion, will be able to have a firm grip over this very important segment of Internet.</p> <p><b>Expected Learning Outcomes :</b></p> <ol style="list-style-type: none"> <li>1. Learner will be able to understand the concepts of networking, which are important for them to be known as a ‘<i>networking professionals</i>’.</li> <li>2. Useful to proceed with industrial requirements and International vendor certifications.</li> </ol>		
<b>Unit I</b>	<p><b>Introduction Network Models:</b></p> <p>Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, standards and administration Internet Standards.</p> <p>Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture, Layers in</p>	<b>15L</b>

	<p>the TCP/IP Protocol Suite, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing. Detailed introduction to Physical Layer, Detailed introduction to Data-Link Layer, Detailed introduction to Network Layer, Detailed introduction to Transport Layer, Detailed introduction to Application Layer.</p> <p>Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay)</p>	
<b>Unit II</b>	<p><b>Introduction to Physical Layer and Data-Link Layer:</b></p> <p>Digital Transmission digital-to-digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM), Transmission Modes, Parallel Transmission, Serial Transmission. Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-to-analog Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM), Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time-Division Multiplexing. Transmission Media, Guided Media, Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching,</p> <p>Introduction to Data-Link Layer, Nodes and Links, Services, Two Sub-layers, Three Types of addresses, Address Resolution Protocol (ARP). Error Detection and Correction, introduction, Types of Errors, Redundancy, Detection versus Correction,</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Network layer, Transport Layer</b></p> <p>Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA.</p> <p>Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer</p>	<b>15L</b>

	<p>Switches, Routers,</p> <p>Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing.</p> <p>Unicast Routing, General Idea, Least-Cost Routing, Routing Algorithms, Distance-Vector Routing, Link-State Routing, Path-Vector Routing, Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection-Oriented Protocols.</p> <p>Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.</p>	
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.</li> <li>2) Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2011.</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Computer Network, Bhushan Trivedi, Oxford University Press</li> <li>2) Data and Computer Communication, William Stallings, PHI</li> </ol>		

<p><b>Course:</b> <b>USCS404</b></p>	<p><b>TOPICS (Credits : 02 Lectures/Week: 03)</b> <b>Software Engineering</b></p>	
<p><b>Unit I</b></p>	<p><b>Introduction:</b> The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming</p> <p><b>Requirement Analysis and System Modeling:</b> Requirements Engineering, Eliciting Requirements, SRS Validation, Components of</p>	<p><b>15L</b></p>



	SRS, Characteristics of SRS , Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram	
<b>Unit II</b>	<p><b>System Design:</b> System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design</p> <p><b>Software Measurement and Metrics:</b> Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality</p> <p><b>Software Project Management: Estimation</b> in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision, <b>Project Scheduling</b> - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Risk Management</b> - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan</p> <p><b>Software Quality Assurance:</b> Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, Software Reliability, The ISO 9000 Quality Standards, Capability Maturity Model</p> <p><b>Software Testing :</b> Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design</p>	<b>15L</b>

**Text book(s):**

- 1) Software Engineering, A Practitioner's Approach, Roger S, Pressman.(2014)

**Additional Reference(s):**

- 1) Software Engineering, Ian Sommerville, Pearson Education
- 2) Software Engineering: Principles and Practices”,Deepak Jain,OXFORD University Press,
- 3) Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI
- 4) Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons
- 5) A Concise Introduction to Software Engineering, Pankaj Jalote, Springer

<b>Course:</b> USCS405	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b> <b>Linear Algebra using Python</b>	
<b>Objectives:</b> To offer the learner the relevant linear algebra concepts through computer science applications.		
<b>Expected Learning Outcomes:</b>		
<ol style="list-style-type: none"> <li>1. Appreciate the relevance of linear algebra in the field of computer science.</li> <li>2. Understand the concepts through program implementation</li> <li>3. Instill a computational thinking while learning linear algebra.</li> </ol>		
<b>Unit I</b>	<b>Field:</b> Introduction to complex numbers, numbers in Python , Abstracting over fields, Playing with GF(2), <b>Vector Space:</b> Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations. Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise	<b>15L</b>
<b>Unit II</b>	<b>Matrix:</b> Matrices as vectors, Transpose, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product,	<b>15L</b>

	<p>From function inverse to matrix inverse</p> <p><b>Basis:</b> Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Linear dependence, Basis , Unique representation, Change of basis, first look, Computational problems involving finding a basis</p> <p>Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator</p>	
<b>Unit III</b>	<p><b>Gaussian elimination:</b> Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers,</p> <p><b>Inner Product:</b> The inner product for vectors over the reals, Orthogonality,</p> <p><b>Orthogonalization:</b> Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement,</p> <p><b>Eigenvector:</b> Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and eigenvectors, Coordinate representation in terms of eigenvectors, The Internet worm, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <p>1) Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)</p> <p><b>Additional References:</b></p> <p>1) Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012).</p> <p>2) Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4<sup>th</sup> Edition (2007).</p> <p>3) Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3<sup>rd</sup> Edition (2002)</p>		

<b>Course:</b> <b>USCS406</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b>  <b>.Net Technologies</b>	
<b>Objectives:</b> To explore .NET technologies for designing and developing dynamic, interactive and responsive web applications.  <b>Expected Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. Understand the .NET framework</li> <li>2. Develop a proficiency in the C# programming language</li> <li>3. Proficiently develop ASP.NET web applications using C#</li> <li>4. Use ADO.NET for data persistence in a web application</li> </ol>		
<b>Unit I</b>	<b>The .NET Framework:</b> .NET Languages, Common Language Runtime, .NET Class Library  <b>C# Language Basics:</b> Comments, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes  <b>ASP.NET:</b> Creating Websites, Anatomy of a Web Form - Page Directive, Doctype, Writing Code - Code-Behind Class, Adding Event Handlers, Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders,  <b>HTML Server Controls</b> - View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File, web.config File	<b>15L</b>
<b>Unit II</b>	<b>Web Controls:</b> Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle  <b>State Management:</b> ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State  <b>Validation:</b> Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions  <b>Rich Controls:</b> Calendar Control, AdRotator Control, MultiView Control  <b>Themes and Master Pages:</b> How Themes Work, Applying a Simple Theme,	<b>15L</b>

	<p>Handling Theme Conflicts, Simple Master Page and Content Page, Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths</p> <p><b>Website Navigation:</b> Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control</p>	
<b>Unit III</b>	<p><b>ADO.NET:</b> Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access</p> <p><b>Data Binding:</b> Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource</p> <p><b>Data Controls:</b> GridView, DetailsView, FormView</p> <p><b>Working with XML:</b> XML Classes – XMLTextWriter, XMLTextReader</p> <p><b>Caching:</b> When to Use Caching, Output Caching, Data Caching</p> <p><b>LINQ:</b> Understanding LINQ, LINQ Basics,</p> <p><b>ASP.NET AJAX:</b> ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <p>1) Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)</p> <p><b>Additional Reference(s):</b></p> <p>1) The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill</p> <p>2) Beginning ASP.NET 4 in C# and VB Imar Spanajaars, WROX</p>		

<b>Course:</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b>	
<b>USCS407</b>	<b>Android Developer Fundamentals</b>	
<p><b>Objectives:</b></p> <p>To provide the comprehensive insight into developing applications running on smart mobile devices and demonstrate programming skills for managing task on mobile. To provide systematic approach for studying definition, methods and its applications for Mobile-App development.</p>		

**Expected Learning Outcomes:**

- 1) Understand the requirements of Mobile programming environment.
- 2) Learn about basic methods, tools and techniques for developing Apps
- 3) Explore and practice App development on Android Platform
- 4) Develop working prototypes of working systems for various uses in daily lives.

<b>Unit I</b>	What is Android? Obtaining the required tools, creating first android app, understanding the components of screen, adapting display orientation, action bar, Activities and Intents, Activity Lifecycle and Saving State, Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, ProgressBar View, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View	<b>15L</b>
<b>Unit II</b>	User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Material design, Providing resources for adaptive layouts, AsyncTask and AsyncTaskLoader, Connecting to the Internet, Broadcast receivers, Services, Notifications, Alarm managers, Transferring data efficiently	<b>15L</b>
<b>Unit III</b>	Data - saving, retrieving, and loading: Overview to storing data, Shared preferences, SQLite primer, store data using SQLite database, ContentProviders, loaders to load and display data, Permissions, performance and security, Firebase and AdMob, Publish your app	<b>15L</b>

**Textbook(s):**

- 1) "Beginning Android 4 Application Development", Wei-Meng Lee, March 2012, WROX.

**Additional Reference(s):**

- 1) <https://developers.google.com/training/courses/android-fundamentals>
- 2) <https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-practicals/details>

## Suggested List of Practical – SEMESTER IV

<b>Course:</b> <b>USCSP401</b>	<b>(Credits : 03 Lectures/Week:09)</b> <b>USCS401+ USCS402+USCS403</b>	
<b>USCS401: Fundamentals of Algorithms</b>		
<ol style="list-style-type: none"> <li>1. Write Python program to perform matrix multiplication. Discuss the complexity of algorithm used.</li> <li>2. Write Python program to sort n names using Quick sort algorithm. Discuss the complexity of algorithm used.</li> <li>3. Write Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of algorithm used.</li> <li>4. Write Python program for inserting an element into binary tree.</li> <li>5. Write Python program for deleting an element (assuming data is given) from binary tree.</li> <li>6. Write Python program for checking whether a given graph G has simple path from source s to destination d. Assume the graph G is represented using adjacent matrix.</li> <li>7. Write Python program for finding the smallest and largest elements in an array A of size n using Selection algorithm. Discuss Time complexity.</li> <li>8. Write Python program for finding the second largest element in an array A of size n using Tournament Method. Discuss Time complexity.</li> <li>9. Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm.</li> <li>10. Write Python program for implementing Strassen's Matrix multiplication using Divide and Conquer method. Discuss the complexity of algorithm.</li> </ol>		
<b>USCS402: Advanced JAVA</b>		
<ol style="list-style-type: none"> <li>1. Develop the presentation layer of Library Management software application with suitable menus.</li> <li>2. Design suitable database for Library Management System.</li> <li>3. Develop business logic layer for Library Management System.</li> <li>4. Develop Java application to store image in a database as well as retrieve image from database.</li> </ol>		

5. Write a Java application to demonstrate servlet life cycle.
6. Design database for student administration. Develop servlet(s) to perform CRUD operations.
7. Create Employees table in EMP database. Perform select, insert, update, and delete operations on Employee table using JSP.
8. Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java application to access JavaBeans Properties.
9. Design application using Struts2. Application must accept user name and greet user when command button is pressed.
10. Write Java application to encoding and decoding JSON in Java.

### **USCS403: Computer Networks**

1. Understanding the working of NIC cards, Ethernet/Fast Ethernet/Gigabit Ethernet.
2. Crimping of Twisted-Pair Cable with RJ45connector for Straight-Through, Cross-Over, Roll-Over.
3. To understand their respective role in networks/internet.
4. Problem solving with IPv4, which will include concept of Classful addressing. (supportive Hint: use Cisco Binary Game)
5. Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: *ping, traceroute, netstat, arp, ipconfig*.
6. Using **Packet Tracer**, create a basic network of two computers using appropriate network wire.
7. Using **Packet Tracer**, connect multiple (min.6) computers using layer 2 switch.
8. Using **Packet Tracer**, connect a network in triangular shape with three layer two switches and every switch will have four computer. Verify their connectivity with each other.
9. Using **Packet Tracer**, create a wireless network of multiple PCs using appropriate access point.
10. Using **Wireshark**, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working.



<b>Course:</b> <b>USCSP402</b>	<b>(Credits : 03 Lectures/Week:09)</b> <b>USCS405+ USCS406+ USCS407</b>	
<b>USCS405: Linear Algebra using Python</b>		
<ol style="list-style-type: none"> <li>1. Write a program which demonstrates the following: <ul style="list-style-type: none"> <li>• Addition of two complex numbers</li> <li>• Displaying the conjugate of a complex number</li> <li>• Plotting a set of complex numbers</li> <li>• Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number <math>a=1/2</math>, <math>a=1/3</math>, <math>a=2</math> etc.</li> </ul> </li> <li>2. Write a program to do the following: <ul style="list-style-type: none"> <li>• Enter a vector <math>u</math> as a <math>n</math>-list</li> <li>• Enter another vector <math>v</math> as a <math>n</math>-list</li> <li>• Find the vector <math>au+bv</math> for different values of <math>a</math> and <math>b</math></li> <li>• Find the dot product of <math>u</math> and <math>v</math></li> </ul> </li> <li>3. Write a program to do the following: <ul style="list-style-type: none"> <li>• Enter two distinct faces as vectors <math>u</math> and <math>v</math>.</li> <li>• Find a new face as a linear combination of <math>u</math> and <math>v</math> i.e. <math>au+bv</math> for <math>a</math> and <math>b</math> in <math>\mathbb{R}</math>.</li> <li>• Find the average face of the original faces.</li> </ul> </li> <li>4. Write a program to do the following: <ul style="list-style-type: none"> <li>• Enter an <math>r</math> by <math>c</math> matrix <math>M</math> (<math>r</math> and <math>c</math> being positive integers)</li> <li>• Display <math>M</math> in matrix format</li> <li>• Display the rows and columns of the matrix <math>M</math></li> <li>• Find the scalar multiplication of <math>M</math> for a given scalar.</li> <li>• Find the transpose of the matrix <math>M</math>.</li> </ul> </li> <li>5. Write a program to do the following: <ul style="list-style-type: none"> <li>• Find the vector-matrix multiplication of a <math>r</math> by <math>c</math> matrix <math>M</math> with an <math>c</math>-vector <math>u</math>.</li> <li>• Find the matrix-matrix product of <math>M</math> with a <math>c</math> by <math>p</math> matrix <math>N</math>.</li> </ul> </li> <li>6. Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse.</li> <li>7. Write a program to convert a matrix into its row echelon form.</li> </ol>		

8. Write a program to do the following:

- Enter a positive number  $N$  and find numbers  $a$  and  $b$  such that  $a^2 - b^2 = N$
- Find the gcd of two numbers using Euclid's algorithm.

9. Write a program to do the following:

- Enter a vector  $b$  and find the projection of  $b$  orthogonal to a given vector  $u$ .
- Find the projection of  $b$  orthogonal to a set of given vectors

10. Write a program to enter a given matrix and an eigen value of the same. Find its eigen vector.

### **USCS406: .NET Technologies**

1. Write C# programs for understanding C# basics involving

- a. Variables and Data Types
- b. Object-Based Manipulation
- c. Conditional Logic
- d. Loops
- e. Methods

2. Write C# programs for Object oriented concepts of C# such as:

- a. Program using classes
- b. Constructor and Function Overloading
- c. Inheritance
- d. Namespaces

3. Design ASP.NET Pages with

- a. Server controls.
- b. Web controls and demonstrate the use of AutoPostBack
- c. Rich Controls (Calendar / Ad Rotator)

4. Design ASP.NET Pages for State Management using

- a. Cookies
- b. Session State
- c. Application State

5. Perform the following activities

- a. Design ASP.NET page and perform validation using various Validation Controls
- b. Design an APS.NET master web page and use it other (at least 2-3) content pages.
- c. Design ASP.NET Pages with various Navigation Controls

6. Performing ADO.NET data access in ASP.NET for

- a. Simple Data Binding
- b. Repeated Value Data Binding

7. Design ASP.NET application for Interacting (Reading / Writing) with XML documents

8. Design ASP.NET Pages for Performance improvement using Caching

9. Design ASP.NET application to query a Database using LINQ

10. Design and use AJAX based ASP.NET pages.

## **USCS407:Android Developer Fundamentals**

1. Install Android Studio and Run Hello World Program.
2. Create an android app with Interactive User Interface using Layouts.
3. Create an android app that demonstrates working with TextView Elements.
4. Create an android app that demonstrates Activity Lifecycle and Instance State.
5. Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers.
6. Create an android app that demonstrates the use of an Options Menu.
7. Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs.
8. Create an android app to Connect to the Internet and use BroadcastReceiver.
9. Create an android app to show Notifications and Alarm manager.
10. Create an android app to save user data in a database and use of different queries.

## Evaluation Scheme

### I. Internal Exam - 25 Marks

#### (i) Test – 20 Marks

20 marks Test – Duration 40 mins

It will be conducted either using any open source learning management system like Moodle (Modular object-oriented dynamic learning environment)

**OR**

A test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.

- (ii) 5 Marks – Active participation in routine class instructional deliveries  
Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

### II. External Exam– 75 Marks

### III. Practical Exam – 50 Marks

- Each course carry 50 Marks : 40 marks + 05 marks (journal) + 05 marks (viva)
- Minimum 75 % practical from each paper are required to be completed and written in the journal.

**(Certified Journal is compulsory for appearing at the time of Practical Exam)**

\*\*\*\*\*

**UNIVERSITY OF MUMBAI**  
No. UG/63 of 2018-19

**CIRCULAR:-**

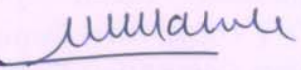
Attention of the Principals of the affiliated Colleges and Directors of the recognized Institutions in Science & Technology Faculty is invited to this office Circular Nos. UG/108 of 2017-18, dated 27<sup>th</sup> July, 2017 relating to syllabus of the Bachelor of Science (B.Sc.) degree course.

They are hereby informed that the recommendations made by the Ad-hoc Board of Studies in Computer Science at its meeting held on 10<sup>th</sup> May, 2018 have been accepted by the Academic Council at its meeting held on 14<sup>th</sup> June, 2018 **vide** item No. 4.40 and that in accordance therewith, the revised syllabus as per the (CBCS) for the T.Y.B.Sc. in Computer Science (Sem - V & VI), has been brought into force with effect from the academic year 2018-19, accordingly. (The same is available on the University's website [www.mu.ac.in](http://www.mu.ac.in)).

MUMBAI - 400 032

6<sup>th</sup> July, 2018

To



(Dr. Dinesh Kamble)  
I/c REGISTRAR

The Principals of the affiliated Colleges & Directors of the recognized Institutions in Science & Technology Faculty. (Circular No. UG/334 of 2017-18 dated 9<sup>th</sup> January, 2018.)

**A.C./4.40/14/06/2018**

\*\*\*\*\*

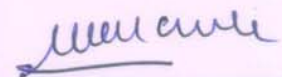
No. UG/ 63 -A of 2018

MUMBAI-400 032

6<sup>th</sup> July, 2018

Copy forwarded with Compliments for information to:-

- 1) The I/c Dean, Faculty of Science & Technology,
- 2) The Chairman, Ad-hoc Board of Studies in Computer Science,
- 3) The Director, Board of Examinations and Evaluation,
- 4) The Director, Board of Students Development,
- 5) The Co-Ordinator, University Computerization Centre,



(Dr. Dinesh Kamble)  
I/c REGISTRAR

Academic Council

Item No: \_\_\_\_\_

# **UNIVERSITY OF MUMBAI**



**Syllabus for Sem V & VI**  
**Program: Bachelor of Science**  
**Course: Computer Science**

Credit Based Semester and Grading System with  
effect from  
Academic Year 2018-2019

## **Preamble**

This is the third year curriculum in the subject of Computer Science. The revised structure is designed to transform students into technically competent, socially responsible and ethical Computer Science professionals. In these Semesters we have made the advancements in the subject based on the previous Semesters Knowledge.

In the first year basic foundation of important skills required for software development is laid. Second year of this course is about studying core computer science subjects. The third year is the further advancement which covers developing capabilities to design formulations of computing models and its applications in diverse areas.

The proposed curriculum contains two semesters, each Semester contains two Electives: Elective-I and II. Every Elective contains three papers based on specific areas of Computer Science. It also includes one Skill Enhancement paper per semester, helps the student to evaluate his/her computer science domain specific skills and also to meet industry expectations. This revised curriculum has not only taken the specific areas of computer science into consideration but will also give the opportunity to the student to prove his/her ability in the subject practically through the Project Implementation. In Semester V and Semester VI student has to undertake a Project. It can boost his/her confidence and also can encourage the student to perform innovations in the subject as the choice of the Project topic is kept open covering most of the areas of Computer Science subject as per the students interest and the subject they have learned during the Course.

Proposed Curriculum contains challenging and varied subjects aligned with the current trend with the introduction of Machine Intelligence specific subject such as Artificial Intelligence, Information Retrieval. Data Management related subjects such as Cloud Computing and Data Science. Image processing topics such as Game Programming, Digital Image Processing. Introduction of physical world through Architecting of IoT and Wireless Sensor Networks and Mobile Communication. Security domain is also evolved by the introduction of Ethical Hacking, Cyber Forensic and Information and Network Security. To get the hands on experience Linux Server Administration and Web Services topics are included.

In essence, the objective of this syllabus is to create a pool of technologically savvy, theoretically strong, innovatively skilled and ethically responsible generation of computer science professionals. Hope that the teacher and student community of University of Mumbai will accept and appreciate the efforts.

**T.Y.B.Sc. (Semester V and VI)**  
**Computer Science Syllabus**  
**Credit Based Semester and Grading System**  
**To be implemented from the Academic year 2018-2019**

<b>SEMESTER V</b>			
<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>L / Week</b>
	<b>Elective-I (Select Any Two)</b>		
USCS501	Artificial Intelligence	3	3
USCS502	Linux Server Administration	3	3
USCS503	Software Testing and Quality Assurance	3	3
	<b>Elective-II (Select Any Two)</b>		
USCS504	Information and Network Security	3	3
USCS505	Architecting of IoT	3	3
USCS506	Web Services	3	3
	<b>Skill Enhancement</b>		
USCS507	Game Programming	2	3
	<b>Practical</b>		
USCSP501	Practical of Elective-I	2	6
USCSP502	Practical of Elective-II	2	6
USCSP503	Project Implementation	1	3
USCSP504	Practical of Skill Enhancement : USCS507	1	3

<b>SEMESTER VI</b>			
<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>L / Week</b>
	<b>Elective-I (Select Any Two)</b>		
USCS601	Wireless Sensor Networks and Mobile Communication	3	3
USCS602	Cloud Computing	3	3
USCS603	Cyber Forensics	3	3
	<b>Elective-II (Select Any Two)</b>		



USCS604	Information Retrieval	3	3
USCS605	Digital Image Processing	3	3
USCS606	Data Science	3	3
	<b>Skill Enhancement</b>		
USCS607	Ethical Hacking	2	3
	<b>Practical</b>		
USCSP601	Practical of Elective-I	2	6
USCSP602	Practical of Elective-II	2	6
USCSP603	Project Implementation	1	3
USCSP604	Practical of Skill Enhancement : USCS607	1	3

## SEMESTER V

### THEORY

<b>Course:</b> <b>USCS501</b>	<b>TOPICS (Credits : 03 Lectures/Week:03)</b> <b>Artificial Intelligence</b>	
<b>Objectives:</b> Artificial Intelligence (AI) and accompanying tools and techniques bring transformational changes in the world. Machines capability to match, and sometimes even surpass human capability, make AI a hot topic in Computer Science. This course aims to introduce the learner to this interesting area.		
<b>Expected Learning Outcomes:</b> After completion of this course, learner should get a clear understanding of AI and different search algorithms used for solving problems. The learner should also get acquainted with different learning algorithms and models used in machine learning.		
<b>Unit I</b>	<b>What Is AI:</b> Foundations, History and State of the Art of AI. <b>Intelligent Agents:</b> Agents and Environments, Nature of Environments, Structure of Agents. <b>Problem Solving by searching:</b> Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.	<b>15L</b>
<b>Unit II</b>	<b>Learning from Examples:</b> Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, Theory of Learning, Regression and Classification with Linear Models, Artificial Neural Networks, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning	<b>15L</b>

<b>Unit III</b>	<b>Learning probabilistic models:</b> Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: The EM Algorithm. Reinforcement learning: Passive Reinforcement Learning, Active Reinforcement Learning, Generalization in Reinforcement Learning, Policy Search, Applications of Reinforcement Learning.	<b>15L</b>
-----------------	--	------------

**Textbook(s):**

- 1) Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010.

**Additional Reference(s):**

- 1) Artificial Intelligence: Foundations of Computational Agents, David L Poole, Alan K. Mackworth, 2nd Edition, Cambridge University Press, 2017.
- 2) Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017
- 3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013

<b>Course:</b> <b>USCS502</b>	<b>TOPICS (Credits : 03 Lectures/Week:03)</b> <b>Linux Server Administration</b>
----------------------------------	---

**Objectives:**

Demonstrate proficiency with the Linux command line interface, directory & file management techniques, file system organization, and tools commonly found on most Linux distributions. Effectively operate a Linux system inside of a network environment to integrate with existing service solutions. Demonstrate the ability to troubleshoot challenging technical problems typically encountered when operating and administering Linux systems.

**Expected Learning Outcomes:**

Learner will be able to develop Linux based systems and maintain. Learner will be able to install appropriate service on Linux server as per requirement. Learner will have proficiency in Linux server administration.

<b>Unit I</b>	<p><b>Introduction:</b> Technical Summary of Linux Distributions, Managing Software</p> <p><b>Single-Host Administration:</b> Managing Users and Groups, Booting and shutting down processes, File Systems, Core System Services, Process of configuring, compiling, Linux Kernel</p> <p><b>Networking and Security:</b> TCP/IP for System Administrators, basic network Configuration, Linux Firewall (Netfilter), System and network security</p>	<b>15L</b>
<b>Unit II</b>	<p><b>Internet Services:</b> Domain Name System (DNS), File Transfer Protocol (FTP), Apache web server, Simple Mail Transfer Protocol (SMTP), Post Office Protocol and Internet Mail Access Protocol (POP and IMAP), Secure Shell (SSH), Network Authentication, OpenLDAP Server, Samba and LDAP, Network authentication system (Kerberos), Domain Name Service (DNS), Security</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Intranet Services:</b> Network File System (NFS), Samba, Distributed File Systems (DFS), Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP), Dynamic Host Configuration Protocol (DHCP), MySQL, LAMP Applications File Servers, Email Services, Chat Applications, Virtual Private Networking.</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Linux Administration: A Beginner's Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016</li> <li>2) Ubuntu Server Guide, Ubuntu Documentation Team, 2016</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016</li> </ol>		

<b>Course:</b> <b>USCS503</b>	<b>TOPICS (Credits : 03 Lectures/Week:03)</b> <b>Software Testing and Quality Assurance</b>	
<b>Objectives:</b> <p>To provide learner with knowledge in Software Testing techniques. To understand how testing methods can be used as an effective tools in providing quality assurance concerning for software.</p> <p>To provide skills to design test case plan for testing software</p> <b>Expected Learning Outcomes:</b> <p>Understand various software testing methods and strategies. Understand a variety of software metrics, and identify defects and managing those defects for improvement in quality for given software. Design SQA activities, SQA strategy, formal technical review report for software quality control and assurance.</p>		
<b>Unit I</b>	<b>Software Testing and Introduction to quality :</b> Introduction, Nature of errors, an example for Testing, Definition of Quality , QA, QC, QM and SQA , Software Development Life Cycle, Software Quality Factors <b>Verification and Validation :</b> Definition of V &V , Different types of V & V Mechanisms, Concepts of Software Reviews, Inspection and Walkthrough <b>Software Testing Techniques :</b> Testing Fundamentals, Test Case Design, White Box Testing and its types, Black Box Testing and its types	<b>15L</b>
<b>Unit II</b>	<b>Software Testing Strategies :</b> Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing <b>Software Metrics :</b> Concept and Developing Metrics, Different types of Metrics, Complexity metrics <b>Defect Management:</b> Definition of Defects, Defect Management Process, Defect Reporting, Metrics Related to Defects, Using Defects for Process Improvement.	<b>15L</b>
<b>Unit III</b>	<b>Software Quality Assurance :</b> Quality Concepts, Quality Movement, Background Issues, SQA activities, Software Reviews, Formal Technical Reviews, Formal approaches to SQA, Statistical Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, , SQA Plan , Six sigma, Informal Reviews	<b>15L</b>

	<p><b>Quality Improvement :</b> Introduction, Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts</p> <p><b>Quality Costs :</b> Defining Quality Costs, Types of Quality Costs, Quality Cost Measurement, Utilizing Quality Costs for Decision-Making</p>	
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4<sup>th</sup> Edition,, Pearson Education, 2005</li> <li>2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 5<sup>th</sup> Edition, Tata McGraw Hill, 2001</li> <li>3. Quality Management, Donna C. S. Summers, 5<sup>th</sup> Edition, Prentice-Hall, 2010.</li> <li>4. Total Quality Management, Dale H. Besterfield, 3<sup>rd</sup> Edition, Prentice Hall, 2003.</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Software engineering: An Engineering approach, J.F. Peters, W. Pedrycz , John Wiley,2004</li> <li>2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley &amp; Sons, Inc. , Publication, 2008</li> <li>3. Software Engineering and Testing, B. B. Agarwal, S. P. Tayal, M. Gupta, Jones and Bartlett Publishers, 2010</li> </ol>		

<p><b>Course:</b> <b>USCS504</b></p>	<p align="center"><b>TOPICS (Credits : 03 Lectures/Week:03)</b> <b>Information and Network Security</b></p>
<p><b>Objectives:</b> To provide students with knowledge of basic concepts of computer security including network security and cryptography.</p> <p><b>Expected Learning Outcomes:</b> Understand the principles and practices of cryptographic techniques. Understand a variety of generic security threats and vulnerabilities, and identify &amp; analyze particular security problems for a given application. Understand various protocols for network security to protect against the threats in a network</p>	

<p><b>Unit I</b></p>	<p><b>Introduction:</b> Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms</p> <p><b>Classical Encryption Techniques:</b> Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers</p> <p><b>Public-Key Cryptography and RSA:</b> Principles of Public-Key Cryptosystems, The RSA Algorithm</p>	<p><b>15L</b></p>
<p><b>Unit II</b></p>	<p><b>Key Management:</b> Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange</p> <p><b>Message Authentication and Hash Functions:</b> Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC</p> <p><b>Digital Signatures and Authentication:</b> Digital Signatures, Authentication Protocols, Digital Signature Standard</p> <p><b>Authentication Applications:</b> Kerberos, X.509 Authentication, Public-Key Infrastructure</p>	<p><b>15L</b></p>
<p><b>Unit III</b></p>	<p><b>Electronic Mail Security:</b> Pretty Good Privacy, S/MIME</p> <p><b>IP Security:</b> Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management</p> <p><b>Web Security:</b> Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction</p> <p><b>Intrusion:</b> Intruders, Intrusion Techniques, Intrusion Detection</p> <p><b>Malicious Software:</b> Viruses and Related Threats, Virus Countermeasures, DDOS</p> <p><b>Firewalls:</b> Firewall Design Principles, Types of Firewalls</p>	<p><b>15L</b></p>
<p><b>Textbook(s):</b></p> <p>1) Cryptography and Network Security: Principles and Practice 5th Edition, William</p>		

Stallings, Pearson,2010

**Additional Reference(s):**

- 1) Cryptography and Network Security, Atul Kahate, Tata McGraw-Hill, 2013.
- 2) Cryptography and Network, Behrouz A Fourouzan, Debdeep Mukhopadhyay, 2<sup>nd</sup> Edition, TMH, 2011

<b>Course:</b> <b>USCS505</b>	<b>TOPICS (Credits : 03 Lectures/Week:03)</b> <b>Architecting of IoT</b>	
<b>Objectives:</b> Discovering the interconnection and integration of the physical world. Learner should get knowledge of the architecture of IoT.		
<b>Expected Learning Outcomes:</b> Learners are able to design & develop IoT Devices. They should also be aware of the evolving world of M2M Communications and IoT analytics.		
<b>Unit I</b>	<b>IoT-An Architectural Overview:</b> Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.  <b>IoT Architecture-State of the Art :</b> Introduction, State of the art, Reference Model and architecture, IoT reference Model - IoT Reference Architecture Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views	<b>15L</b>
<b>Unit II</b>	<b>IoT Data Link Layer and Network Layer Protocols:</b>  PHY/MAC Layer(3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART,Z-Wave, Bluetooth Low Energy, Zigbee Smart Energy DASH7  <b>Network Layer:</b> IPv4, IPv6, 6LoWPAN, 6TiSCH,ND, DHCP, ICMP, RPL, CORPL, CARP	<b>15L</b>



<b>Unit III</b>	<p><b>Transport layer protocols :</b> Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP)-(TLS, DTLS)</p> <p><b>Session layer:</b> Session Layer-HTTP, CoAP, XMPP, AMQP, MQTT</p> <p><b>Service layer protocols:</b> Service Layer -oneM2M, ETSI M2M, OMA, BBF</p>	<b>15L</b>
-----------------	--	------------

**Textbook(s):**

1. From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, 1st Edition, Academic Press, 2014.
2. Learning Internet of Things, Peter Waher, PACKT publishing, BIRMINGHAM – MUMBAI, 2015

**Additional References(s):**

1. Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, Daniel Minoli, Wiley Publications, 2013
2. Internet of Things (A Hands-on Approach), Vijay Madisetti and Arshdeep Bahga, 1st Edition, VPT, 2014.
3. [http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot\\_prot/index.html](http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html)

<b>Course:</b> <b>USCS506</b>	<b>TOPICS (Credits : 03 Lectures/Week:03)</b> <b>Web Services</b>
<p><b>Objectives:</b> To understand the details of web services technologies like SOAP, WSDL, and UDDI. To learn how to implement and deploy web service client and server. To understand the design principles and application of SOAP and REST based web services (JAX-WS and JAX-RS). To understand WCF service. To design secure web services and QoS of Web Services</p> <p><b>Expected Learning Outcomes:</b> Emphasis on SOAP based web services and associated standards such as WSDL. Design SOAP based / RESTful / WCF services Deal with Security and QoS issues of Web Services</p>	

<b>Unit I</b>	<p><b>Web services basics :</b></p> <p>What Are Web Services? Types of Web Services Distributed computing infrastructure, overview of XML, SOAP, Building Web Services with JAX-WS, Registering and Discovering Web Services, Service Oriented Architecture, Web Services Development Life Cycle, Developing and consuming simple Web Services across platform</p>	<b>15L</b>
<b>Unit II</b>	<p><b>The REST Architectural style :</b></p> <p>Introducing HTTP, The core architectural elements of a RESTful system, Description and discovery of RESTful web services, Java tools and frameworks for building RESTful web services, JSON message format and tools and frameworks around JSON, Build RESTful web services with JAX-RS APIs, The Description and Discovery of RESTful Web Services, Design guidelines for building RESTful web services, Secure RESTful web services</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Developing Service-Oriented Applications with WCF :</b></p> <p>What Is Windows Communication Foundation, Fundamental Windows Communication Foundation Concepts, Windows Communication Foundation Architecture, WCF and .NET Framework Client Profile, Basic WCF Programming, WCF Feature Details. Web Service QoS</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Web Services: Principles and Technology, Michael P. Papazoglou, Pearson Education Limited, 2008</li> <li>2) RESTful Java Web Services, Jobinesh Purushothaman, PACKT Publishing, 2<sup>nd</sup> Edition, 2015</li> <li>3) Developing Service-Oriented Applications with WCF, Microsoft, 2017 <a href="https://docs.microsoft.com/en-us/dotnet/framework/wcf/index">https://docs.microsoft.com/en-us/dotnet/framework/wcf/index</a></li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Leonard Richardson and Sam Ruby, RESTful Web Services, O'Reilly, 2007</li> <li>2) The Java EE 6Tutorial, Oracle, 2013</li> </ol>		

<b>Course:</b> <b>USCS507</b>	<b>TOPICS (Credits : 03 Lectures/Week: 03)</b>  <b>Game Programming</b>	
<b>Objectives:</b> Learner should get the understanding computer Graphics programming using Directx or Opengl. Along with the VR and AR they should also aware of GPU, newer technologies and programming using most important API for windows. <b>Expected Learning Outcomes:</b> Learner should study Graphics and gaming concepts with present working style of developers where everything remains on internet and they need to review it, understand it, be a part of community and learn.		
<b>Unit I</b>	<b>Mathematics for Computer Graphics, DirectX Kickstart:</b>  <b>Cartesian Coordinate system:</b> The Cartesian XY-plane, Function Graphs, Geometric Shapes, Polygonal Shapes, Areas of Shapes, Theorem of Pythagoras in 2D, Coordinates, Theorem of Pythagoras in 3D, 3D Polygons, Euler’s Rule  <b>Vectors:</b> Vector Manipulation, multiplying a Vector by a Scalar, Vector Addition and Subtraction, Position Vectors, Unit Vectors, Cartesian Vectors, Vector Multiplication, Scalar Product, Example of the Dot Product, The Dot Product in Lighting Calculations, The Dot Product in Back-Face Detection, The Vector Product, The Right-Hand Rule, deriving a Unit Normal Vector for a Triangle Areas, Calculating 2D Areas  <b>Transformations:</b> 2D Transformations, Matrices, Homogeneous Coordinates, 3D Transformations, Change of Axes, Direction Cosines, rotating a Point about an Arbitrary Axis, Transforming Vectors, Determinants, Perspective Projection, Interpolation  <b>DirectX:</b> Understanding GPU and GPU architectures. How they are different from CPU Architectures? Understanding how to solve by GPU?	<b>15L</b>

<p><b>Unit II</b></p>	<p><b>DirectX Pipeline and Programming:</b>  <b>Introduction To DirectX 11:</b> COM, Textures and Resources Formats, The swap chain and Page flipping, Depth Buffering, Texture Resource Views, Multisampling Theory and MS in Direct3D, Feature Levels  <b>Direct3D 11 Rendering Pipeline:</b> Overview, Input Assembler Stage (IA), Vertex Shader Stage (VS), The Tessellation Stage (TS), Geometry Shader Stage (GS), Pixel Shader Stage (PS), Output merger Stage (OM)  Understanding Meshes or Objects, Texturing, Lighting, Blending.  <b>Interpolation and Character Animation:</b>  <b>Trigonometry:</b> The Trigonometric Ratios, Inverse Trigonometric Ratios, Trigonometric Relationships, The Sine Rule, The Cosine Rule, Compound Angles, Perimeter Relationships  <b>Interpolation:</b> Linear Interpolant, Non-Linear Interpolation, Trigonometric Interpolation, Cubic Interpolation, Interpolating Vectors, Interpolating Quaternions  <b>Curves:</b> Circle, Bezier, B-Splines  <b>Analytic Geometry:</b> Review of Geometry, 2D Analytic Geometry, Intersection Points, Point in Triangle, and Intersection of circle with straight line.</p>	<p><b>15L</b></p>
<p><b>Unit III</b></p>	<p><b>Introduction to Rendering Engines:</b> Understanding the current market Rendering Engines. Understanding AR, VR and MR. Depth Mappers, Mobile Phones, Smart Glasses, HMD's  <b>Unity Engine: Multi-platform publishing, VR + AR:</b> Introduction and working in Unity, 2D, Graphics, Physics, Scripting, Animation, Timeline, Multiplayer and Networking, UI, Navigation and Pathfinding, XR, Publishing.  <b>Scripting:</b> Scripting Overview, Scripting Tools and Event Overview  <b>XR:</b> VR, AR, MR, Conceptual Differences. SDK, Devices</p>	<p><b>15L</b></p>
<p><b>Text Book(s):</b></p> <ol style="list-style-type: none"> <li>1) Mathematics for Computer Graphics, John Vince, Springer-Verlag London, 5<sup>th</sup> Edition, 2017</li> <li>2) Mathematics for 3D Game Programming and Computer Graphic, Eric Lengyel, Delmar</li> </ol>		

Cengage Learning, Delmar Cengage Learning,2011

- 3) Introduction To 3D Game Programming With DirectX® 11, Frank D Luna, Mercury Learning And Information,2012.
- 4) <https://docs.unity3d.com/Manual/index.html> - Free

**Additional Reference(s):**

- 1) Computer Graphics, C Version, Donald Hern and Pauline Baker, Pearson Education, 2<sup>nd</sup> Edition, 1997
- 2) HLSL Development Cookbook, Doron Feinstein, PACKT Publishing,2013

## Suggested List of Practical- SEMESTER V

<b>Course:</b> <b>USCSP501</b>	<b>(Credits : 02 Lectures/Week: 06)</b> <b>Practical of Elective-I</b>	
<b>USCS501: Artificial Intelligence</b>		
<p style="text-align: center;"><i>Practical shall be implemented in LISP</i></p> <ol style="list-style-type: none"> <li>1. Implement Breadth first search algorithm for Romanian map problem.</li> <li>2. Implement Iterative deep depth first search for Romanian map problem.</li> <li>3. Implement A* search algorithm for Romanian map problem.</li> <li>4. Implement recursive best-first search algorithm for Romanian map problem.</li> <li>5. Implement decision tree learning algorithm for the restaurant waiting problem.</li> <li>6. Implement feed forward back propagation neural network learning algorithm for the restaurant waiting problem.</li> <li>7. Implement Adaboost ensemble learning algorithm for the restaurant waiting problem.</li> <li>8. Implement Naive Bayes' learning algorithm for the restaurant waiting problem.</li> <li>9. Implement passive reinforcement learning algorithm based on adaptive dynamic programming (ADP) for the 3 by 4 world problem</li> <li>10. Implement passive reinforcement learning algorithm based on temporal differences (TD) for 3 by 4 world problem.</li> </ol>		
<b>USCS502: Linux Server Administration</b>		
<p><i>- Practical shall be performed using any Linux Server (with 8GB RAM).</i></p> <p><i>- Internet connection will be required so that Linux server (command line mode) can be connected to Internet.</i></p> <ol style="list-style-type: none"> <li>1. Install DHCP Server in Ubuntu 16.04</li> <li>2. Initial settings: Add a User, Network Settings, Change to static IP address, Disable IPv6 if not needed, Configure Services, display the list of services which are running, Stop and turn OFF auto-start setting for a service if you don't need it, Sudo Settings</li> <li>3. Configure NTP Server (NTPd), Install and Configure NTPd, Configure NTP Client (Ubuntu and Windows)</li> <li>4. SSH Server : Password Authentication</li> </ol>		

Configure SSH Server to manage a server from the remote computer, SSH Client : (Ubuntu and Windows)

5. Install DNS Server BIND, Configure DNS server which resolves domain name or IP address, Install BIND 9, Configure BIND, Limit ranges you allow to access if needed.
6. Configure DHCP Server, Configure DHCP (Dynamic Host Configuration Protocol) Server, Configure NFS Server to share directories on your Network, Configure NFS Client. (Ubuntu and Windows Client OS)
7. Configure LDAP Server, Configure LDAP Server in order to share users' accounts in your local networks, Add LDAP User Accounts in the OpenLDAP Server, Configure LDAP Client in order to share users' accounts in your local networks. Install phpLDAPadmin to operate LDAP server via Web browser.
8. Configure NIS Server in order to share users' accounts in your local networks, Configure NIS Client to bind NIS Server.
9. Install MySQL to configure database server, Install phpMyAdmin to operate MySQL on web browser from Clients.
10. Install Samba to share folders or files between Windows and Linux.

### **USCS503: Software Testing and Quality Assurance**

1. Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.
2. Conduct a test suite for any two web sites.
3. Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP.
4. Write and test a program to login a specific web page.
5. Write and test a program to update 10 student records into table into Excel file
6. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).
7. Write and test a program to provide total number of objects present / available on the page.
8. Write and test a program to get the number of items in a list / combo box.
9. Write and test a program to count the number of check boxes on the page checked and unchecked count.
10. Load Testing using JMeter, Android Application testing using Appium Tools, Bugzilla Bug tracking tools.

<b>Course:</b> <b>USCSP502</b>	<b>(Credits : 02 Lectures/Week: 06)</b> <b>Practical of Elective-II</b>	
<b>USCS504: Information and Network security</b>		
<ol style="list-style-type: none"> <li>1. Write programs to implement the following Substitution Cipher Techniques: <ul style="list-style-type: none"> <li>- Caesar Cipher</li> <li>- Monoalphabetic Cipher</li> </ul> </li> <li>2 Write programs to implement the following Substitution Cipher Techniques: <ul style="list-style-type: none"> <li>- Vernam Cipher</li> <li>- Playfair Cipher</li> </ul> </li> <li>3 Write programs to implement the following Transposition Cipher Techniques: <ul style="list-style-type: none"> <li>- Rail Fence Cipher</li> <li>- Simple Columnar Technique</li> </ul> </li> <li>4 Write program to encrypt and decrypt strings using <ul style="list-style-type: none"> <li>- DES Algorithm</li> <li>- AES Algorithm</li> </ul> </li> <li>5 Write a program to implement RSA algorithm to perform encryption / decryption of a given string.</li> <li>6 Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.</li> <li>7 Write a program to implement the MD5 algorithm compute the message digest.</li> <li>8 Write a program to calculate HMAC-SHA1 Signature</li> <li>9 Write a program to implement SSL.</li> <li>10 Configure Windows Firewall to block: <ul style="list-style-type: none"> <li>- A port</li> <li>- An Program</li> <li>- A website</li> </ul> </li> </ol>		
<b>USCS505: Architecting of IoT</b>		
<ol style="list-style-type: none"> <li>1. a) Edit text files with nano and cat editor, Learn sudo privileges and Unix shell commands such as cd , ls , cat, etc</li> </ol>		



b) Learn to set dynamic and static IP. Connect to an Ethernet and WiFi network.

Learn to vnc and ssh into a raspberry pi using vnc and putty from a different computer on the network.

c) Write a basic bash script to open programs in kiosk mode. Learn how to autostart programs on boot.

2. Run the node red editor and run simple programs and trigger gpios. Use basic nodes such as inject, debug, gpio

3. Open the python idle editor and run simple Python scripts such as to print Fibonacci numbers, string functions. Learn how to install modules using Pip and write functions

4. Setup a physical button switch and trigger an led in node red and python w debounce

5. Write simple JavaScript functions in Node-Red simple HTTP server page using node red

6. Setup a TCP server and client on a raspberry pi using Python modules to send messages and execute shell commands from within python such as starting another application

7. Trigger a set of led Gpios on the pi via a Python Flask web server

8. Interface the raspberry pi with a 16x2 LCD display and print values.

9. Setup a Mosquitto MQTT server and client and write a Python script to communicate data between Pi's.

10. Interface with an Accelerometer Gyro Mpu6050 on the i2c bus and send sensor values over the internet via mqtt.

### **USCS506: Web Services**

1. Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and vice versa.

2. Write a program to implement the operation can receive request and will return a response in two ways. a) One - Way operation b) Request –Response

3. Write a program to implement business UDDI Registry entry.

4. Develop client which consumes web services developed in different platform.

5. Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.

6. Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.

7. Define a RESTful web service that accepts the details to be stored in a database and performs

CRUD operation.

8. Implement a typical service and a typical client using WCF.
9. Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service.
10. Demonstrates using the binding attribute of an endpoint element in WCF.

**Course:**  
**USCSP503**

**(Credits : 01 Lectures/Week: 03)**  
**Project Implementation**

*Please Refer to Project Implementation Guidelines*

**Course:**  
**USCSP504**

**(Credits : 01 Lectures/Week: 03)**  
**Practical of Skill Enhancement**

**USCS507 : Game Programming**

1. Setup DirectX 11, Window Framework and Initialize Direct3D Device
2. Buffers, Shaders and HLSL (Draw a triangle using Direct3D 11)
3. Texturing (Texture the Triangle using Direct 3D 11)
4. Lightning (Programmable Diffuse Lightning using Direct3D 11)
5. Specular Lightning (Programmable Spot Lightning using Direct3D 11)
6. Loading models into DirectX 11 and rendering.

**Perform following Practical using online content from the Unity Tutorials Web--sites:**

**<https://unity3d.com/learn/tutorials/s/interactive-tutorials>**

7. <https://unity3d.com/learn/tutorials/s/2d-ufo-tutorial>
8. <https://unity3d.com/learn/tutorials/s/space-shooter-tutorial>
9. <https://unity3d.com/learn/tutorials/s/roll-ball-tutorial>
10. <https://unity3d.com/learn/tutorials/topics/vr/introduction?playlist=22946>

## SEMESTER VI

### THEORY

<b>Course:</b> USCS601	<b>TOPICS (Credits : 03 Lectures/Week: 03)</b> <b>Wireless Sensor Networks and Mobile Communication</b>	
<b>Objectives:</b> In this era of wireless and adhoc network, connecting different wireless devices and understanding their compatibility is very important. Information is gathered in many different ways from these devices. Learner should be able to conceptualize and understand the framework. On completion, will be able to have a firm grip over this very important segment of wireless network.		
<b>Expected Learning Outcomes:</b> After completion of this course, learner should be able to list various applications of wireless sensor networks, describe the concepts, protocols, design, implementation and use of wireless sensor networks. Also implement and evaluate new ideas for solving wireless sensor network design issues.		
<b>Unit I</b>	<b>Introduction:</b> Introduction to Sensor Networks, unique constraints and challenges. Advantage of Sensor Networks, Applications of Sensor Networks, Mobile Adhoc NETWORKS (MANETs) and Wireless Sensor Networks, Enabling technologies for Wireless Sensor Networks. <b>Sensor Node Hardware and Network Architecture:</b> Single-node architecture, Hardware components & design constraints, Operating systems and execution environments, introduction to TinyOS and nesC. Network architecture, Optimization goals and figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts.	<b>15L</b>
<b>Unit II</b>	<b>Medium Access Control Protocols:</b> Fundamentals of MAC Protocols, MAC Protocols for WSNs, Sensor-MAC Case Study. <b>Routing Protocols :</b> Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless Sensor Networks, Routing Strategies in Wireless Sensor Networks. <b>Transport Control Protocols :</b> Traditional Transport Control Protocols,	<b>15L</b>

	Transport Protocol Design Issues, Examples of Existing Transport Control Protocols, Performance of Transport Control Protocols.	
<b>Unit III</b>	<p><b>Introduction, Wireless Transmission and Medium Access Control:</b> Applications, A short history of wireless communication.</p> <p><b>Wireless Transmission:</b> Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems.</p> <p><b>Telecommunication, Satellite and Broadcast Systems: GSM:</b> Mobile services, System architecture, Radio interface, Protocols, Localization And Calling, Handover, security, New data services; DECT: System architecture, Protocol architecture; ETRA, UMTS and IMT- 2000.</p> <p>Satellite Systems: History, Applications, Basics: GEO, LEO, MEO; Routing, Localization, Handover.</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2005</li> <li>2) Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley &amp; Sons, 2007</li> <li>3) Mobile communications, Jochen Schiller,2<sup>nd</sup> Edition, Addison wisely , Pearson Education,2012</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Fundamentals of Wireless Sensor Networks, Theory and Practice, Waltenequs Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011</li> <li>2) Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005</li> </ol>		

<b>Course:</b> <b>USCS602</b>	<b>TOPICS (Credits : 03 Lectures/Week: 03)</b>  <b>Cloud Computing</b>	
<p><b>Objectives:</b></p> <p>To provide learners with the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture, implantations and applications. To expose the learners to frontier areas of Cloud Computing, while providing sufficient foundations to enable further study and research.</p> <p><b>Expected Learning Outcomes:</b></p> <p>After successfully completion of this course, learner should be able to articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing using open source technology. Learner should be able to identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc. They should explain the core issues of cloud computing such as security, privacy, and interoperability.</p>		
<b>Unit I</b>	Introduction to Cloud Computing, Characteristics and benefits of Cloud Computing, Basic concepts of Distributed Systems, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing. Elements of Parallel Computing. Elements of Distributed Computing. Technologies for Distributed Computing. Cloud Computing Architecture. The cloud reference model. Infrastructure as a service. Platform as a service. Software as a service. Types of clouds.	<b>15L</b>
<b>Unit II</b>	Characteristics of Virtualized Environments. Taxonomy of Virtualization Techniques. Virtualization and Cloud Computing. Pros and Cons of Virtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment. Open challenges of Cloud Computing	<b>15L</b>
<b>Unit III</b>	Introduction to OpenStack, OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat	<b>15L</b>

**Textbook(s):**

- 1) Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013
- 2) OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016

**Additional Reference(s):**

- 1) OpenStack Essentials, Dan Radez, PACKT Publishing, 2015
- 2) OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O'Reilly Media, Inc., 2014
- 3) <https://www.openstack.org>

**Course:****TOPICS (Credits :03 Lectures/Week:03)****USCS603****Cyber Forensics****Objectives:**

To understand the procedures for identification, preservation, and extraction of electronic evidence, auditing and investigation of network and host system intrusions, analysis and documentation of information gathered

**Expected Learning Outcomes :**

The student will be able to plan and prepare for all stages of an investigation - detection, initial response and management interaction, investigate various media to collect evidence, report them in a way that would be acceptable in the court of law.

**Unit I****Computer Forensics :**

Introduction to Computer Forensics and standard procedure, Incident Verification and System Identification ,Recovery of Erased and damaged data, Disk Imaging and Preservation, Data Encryption and Compression, Automated Search Techniques, Forensics Software

**Network Forensic :**

Introduction to Network Forensics and tracking network traffic, Reviewing Network Logs, Network Forensics Tools, Performing Live Acquisitions, Order of Volatility, Standard Procedure

**Cell Phone and Mobile Device Forensics:** Overview, Acquisition Procedures for Cell Phones and Mobile Devices

**15L**

<p><b>Unit II</b></p>	<p><b>Internet Forensic :</b>  <b>Introduction</b> to Internet Forensics, World Wide Web Threats, Hacking and Illegal access, Obscene and Incident transmission, Domain Name Ownership Investigation, Reconstructing past internet activities and events  <b>E-mail Forensics :</b> e-mail analysis, e-mail headers and spoofing, Laws against e-mail Crime, <b>Messenger Forensics:</b> Yahoo Messenger  <b>Social Media Forensics:</b> Social Media Investigations  <b>Browser Forensics:</b> Cookie Storage and Analysis, Analyzing Cache and temporary internet files, Web browsing activity reconstruction</p>	<p><b>15L</b></p>
<p><b>Unit III</b></p>	<p><b>Investigation, Evidence presentation and Legal aspects of Digital Forensics:</b> Authorization to collect the evidence , Acquisition of Evidence, Authentication of the evidence, Analysis of the evidence, Reporting on the findings, Testimony  <b>Introduction to Legal aspects of Digital Forensics:</b> Laws &amp; regulations, Information Technology Act, Giving Evidence in court, Case Study – Cyber Crime cases, Case Study – Cyber Crime cases</p>	<p><b>15L</b></p>

**Textbook(s):**

1. Guide to computer forensics and investigations, Bill Nelson, Amelia Philips and Christopher Steuart, course technology,5th Edition,2015

**Additional Reference(s):**

2. Incident Response and computer forensics, Kevin Mandia, Chris Prorise, Tata McGrawHill,2<sup>nd</sup> Edition,2003

<p><b>Course:</b> USCS604</p>	<p><b>TOPICS (Credits : 03 Lectures/Week: 03)</b> <b>Information Retrieval</b></p>	
<p><b>Objectives:</b> To provide an overview of the important issues in classical and web information retrieval. The focus is to give an up-to- date treatment of all aspects of the design and implementation of systems for gathering, indexing, and searching documents and of methods for evaluating systems.</p> <p><b>Expected Learning Outcomes:</b></p>		

After completion of this course, learner should get an understanding of the field of information retrieval and its relationship to search engines. It will give the learner an understanding to apply information retrieval models.

<b>Unit I</b>	<b>Introduction to Information Retrieval:</b> Introduction, History of IR, Components of IR, and Issues related to IR, Boolean retrieval, Dictionaries and tolerant retrieval.	<b>15L</b>
<b>Unit II</b>	<b>Link Analysis and Specialized Search:</b> Link Analysis, hubs and authorities, Page Rank and HITS algorithms, Similarity, Hadoop & Map Reduce, Evaluation, Personalized search, Collaborative filtering and content-based recommendation of documents and products, handling “invisible” Web, Snippet generation, Summarization, Question Answering, Cross- Lingual Retrieval.	<b>15L</b>
<b>Unit III</b>	<b>Web Search Engine:</b> Web search overview, web structure, the user, paid placement, search engine optimization/spam, Web size measurement, search engine optimization/spam, Web Search Architectures. <b>XML retrieval:</b> Basic XML concepts, Challenges in XML retrieval, A vector space model for XML retrieval, Evaluation of XML retrieval, Text-centric versus data-centric XML retrieval.	<b>15L</b>

**Text book(s):**

- 1) Introduction to Information Retrieval, C. Manning, P. Raghavan, and H. Schütze, Cambridge University Press, 2008
- 2) Modern Information Retrieval: The Concepts and Technology behind Search, Ricardo Baeza -Yates and Berthier Ribeiro – Neto, 2<sup>nd</sup> Edition, ACM Press Books 2011.
- 3) Search Engines: Information Retrieval in Practice, Bruce Croft, Donald Metzler and Trevor Strohman, 1<sup>st</sup> Edition, Pearson, 2009.

**Additional Reference(s):**

- 1) Information Retrieval Implementing and Evaluating Search Engines, Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, The MIT Press; Reprint edition (February 12, 2016)



<b>Course:</b> <b>USCS605</b>	<b>TOPICS (Credits : 03 Lectures/Week: 03)</b>  <b>Digital Image Processing</b>	
<p><b>Objectives:</b>  To study two-dimensional Signals and Systems. To understand image fundamentals and transforms necessary for image processing. To study the image enhancement techniques in spatial and frequency domain. To study image segmentation and image compression techniques.</p> <p><b>Expected Learning Outcomes:</b>  Learner should review the fundamental concepts of a digital image processing system. Analyze the images in the frequency domain using various transforms. Evaluate the techniques for image enhancement and image segmentation. Apply various compression techniques. They will be familiar with basic image processing techniques for solving real problems.</p>		
<b>Unit I</b>	<p><b>Introduction to Image-processing System :</b> Introduction, Image Sampling, Quantization, Resolution, Human Visual Systems, Elements of an Image-processing System, Applications of Digital Image Processing</p> <p><b>2D Signals and Systems :</b> 2D signals, separable sequence, periodic sequence, 2D systems, classification of 2D systems, 2D Digital filter</p> <p><b>Convolution and Correlation :</b> 2D Convolution through graphical method, Convolution through 2D Z—transform, 2D Convolution through matrix analysis, Circular Convolution, Applications of Circular Convolution, 2D Correlation</p> <p><b>Image Transforms:</b> Need for transform, image transforms, Fourier transform, 2D Discrete Fourier Transform, Properties of 2D DFT, Importance of Phase, Walsh transform, Hadamard transform, Haar transform, Slant transform, Discrete Cosine transform, KL transform</p>	<b>15L</b>
<b>Unit II</b>	<p><b>Image Enhancement :</b>Image Enhancement in spatial domain, Enhancement through Point operations, Histogram manipulation, Linear and nonlinear Gray Level Transformation, local or neighborhood operation, Median Filter, Spatial domain High pass filtering, Bit-plane slicing, Image Enhancement in frequency domain, Homomorphic filter, Zooming operation, Image Arithmetic</p>	<b>15L</b>

	<p><b>Binary Image processing</b> :Mathematical morphology, Structuring elements, Morphological image processing, Logical operations, Morphological operations, Dilation and Erosion, Distance Transform</p> <p><b>Colour Image processing</b> :Colour images, Colour Model, Colour image quantization, Histogram of a colour image</p>	
<b>Unit III</b>	<p><b>Image Segmentation:</b> Image segmentation techniques, Region approach, Clustering techniques, Thresholding, Edge-based segmentation, Edge detection, Edge Linking, Hough Transform</p> <p><b>Image Compression:</b> Need for image compression, Redundancy in images, Image-compression scheme, Fundamentals of Information Theory, Run-length coding, Shannon-Fano coding, Huffman Coding, Arithmetic Coding, Transform-based compression, Image-compression standard</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <p>1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009</p> <p><b>Additional Reference(s):</b></p> <p>1) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008</p> <p>2) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan And T. Veerakumar, 2016 (<a href="https://scilab.in/textbook_companion/generate_book/125">https://scilab.in/textbook_companion/generate_book/125</a>)</p>		
<b>Course:</b> <b>USCS606</b>	<p><b>TOPICS (Credits : 03 Lectures/Week: 03)</b></p> <p><b>Data Science</b></p>	
<p><b>Objectives:</b></p> <p>Understanding basic data science concepts. Learning to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization. Making aware of how to address advanced statistical situations, Modeling and Machine Learning.</p> <p><b>Expected Learning Outcomes:</b></p> <p>After completion of this course, the students should be able to understand &amp; comprehend the problem; and should be able to define suitable statistical method to be adopted.</p>		
<b>Unit I</b>	<b>Introduction to Data Science:</b> What is Data? Different kinds of data,	<b>15L</b>

	<p>Introduction to high level programming language + Integrated Development Environment (IDE), Exploratory Data Analysis (EDA) + Data Visualization, Different types of data sources,</p> <p><b>Data Management:</b> Data Collection, Data cleaning/extraction, Data analysis &amp; Modeling</p>	
<b>Unit II</b>	<p><b>Data Curation:</b> Query languages and Operations to specify and transform data, Structured/schema based systems as users and acquirers of data</p> <p>Semi-structured systems as users and acquirers of data, Unstructured systems in the acquisition and structuring of data, Security and ethical considerations in relation to authenticating and authorizing access to data on remote systems, Software development tools, Large scale data systems, Amazon Web Services (AWS)</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Statistical Modelling and Machine Learning:</b></p> <p>Introduction to model selection: Regularization, bias/variance tradeoff e.g. parsimony, AIC, BIC, Cross validation, Ridge regressions and penalized regression e.g. LASSO</p> <p><b>Data transformations:</b> Dimension reduction, Feature extraction, Smoothing and aggregating</p> <p><b>Supervised Learning:</b> Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN</p> <p><b>Unsupervised Learning:</b> Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Ensemble methods</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Doing Data Science, Rachel Schutt and Cathy O’Neil, O’Reilly,2013</li> <li>2) Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Hands-On Programming with R, Garrett Golemund,1<sup>st</sup> Edition, 2014</li> <li>2) An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015</li> </ol>		

<b>Course:</b> <b>USCS607</b>	<b>TOPICS (Credits : 02 Lectures/Week: 03)</b>  <b>Ethical Hacking</b>	
<b>Objectives:</b> To understand the ethics, legality, methodologies and techniques of hacking.  <b>Expected Learning Outcomes:</b> Learner will know to identify security vulnerabilities and weaknesses in the target applications. They will also know to test and exploit systems using various tools and understand the impact of hacking in real time machines.		
<b>Unit I</b>	<b>Information Security : Attacks and Vulnerabilities</b> <b>Introduction to information security :</b> Asset, Access Control, CIA, Authentication, Authorization, Risk, Threat, Vulnerability, Attack, Attack Surface, Malware, Security-Functionality-Ease of Use Triangle <b>Types of malware :</b> Worms, viruses, Trojans, Spyware, Rootkits <b>Types of vulnerabilities :</b> OWASP Top 10 : cross-site scripting (XSS), cross site request forgery (CSRF/XSRF), SQL injection, input parameter manipulation, broken authentication, sensitive information disclosure, XML External Entities, Broken access control, Security Misconfiguration, Using components with known vulnerabilities, Insufficient Logging and monitoring, OWASP Mobile Top 10, CVE Database <b>Types of attacks and their common prevention mechanisms :</b> Keystroke Logging, Denial of Service (DoS /DDoS), Waterhole attack, brute force, phishing and fake WAP, Eavesdropping, Man-in-the-middle, Session Hijacking, Clickjacking, Cookie Theft, URL Obfuscation, buffer overflow, DNS poisoning, ARP poisoning, Identity Theft, IoT Attacks, BOTs and BOTNETs <b>Case-studies :</b> Recent attacks – Yahoo, Adult Friend Finder, eBay, Equifax, WannaCry, Target Stores, Uber, JP Morgan Chase, Bad Rabbit	<b>15L</b>
<b>Unit II</b>	<b>Ethical Hacking – I (Introduction and pre-attack)</b> <b>Introduction:</b> Black Hat vs. Gray Hat vs. White Hat (Ethical) hacking, Why is Ethical hacking needed?, How is Ethical hacking different from security auditing and digital forensics?, Signing NDA, Compliance and Regulatory	<b>15L</b>

	<p>concerns, Black box vs. White box vs. Black box, Vulnerability assessment and Penetration Testing.</p> <p><b>Approach : Planning</b> - Threat Modeling, set up security verification standards, Set up security testing plan – When, which systems/apps, understanding functionality, black/gray/white, authenticated vs. unauthenticated, internal vs. external PT, Information gathering, Perform Manual and automated (Tools: WebInspect/Qualys, Nessus, Proxies, Metasploit) VA and PT, How WebInspect/Qualys tools work: Crawling/Spidering, requests forging, pattern matching to known vulnerability database and Analyzing results, Preparing report, Fixing security gaps following the report</p> <p><b>Enterprise strategy</b> : Repeated PT, approval by security testing team, Continuous Application Security Testing,</p> <p><b>Phases:</b> Reconnaissance/foot-printing/Enumeration, <b>Phases:</b> Scanning, Sniffing</p>	
<p><b>Unit III</b></p>	<p><b>Ethical Hacking :Enterprise Security</b></p> <p><b>Phases : Gaining and Maintaining Access : Systems hacking</b> – Windows and Linux – Metasploit and Kali Linux, Keylogging, Buffer Overflows, Privilege Escalation, Network hacking - ARP Poisoning, Password Cracking, WEP Vulnerabilities, MAC Spoofing, MAC Flooding, IPSpoofing, SYN Flooding, Smurf attack, <b>Applications hacking</b> : SMTP/Email-based attacks, VOIP vulnerabilities, Directory traversal, Input Manipulation, Brute force attack, Unsecured login mechanisms, SQL injection, XSS, Mobile apps security, <b>Malware analysis</b> : Netcat Trojan, wrapping definition, reverse engineering</p> <p><b>Phases</b> : Covering your tracks : Steganography, Event Logs alteration</p> <p><b>Additional Security Mechanisms</b> : IDS/IPS, Honeypots and evasion techniques, Secure Code Reviews (Fortify tool, OWASP Secure Coding Guidelines)</p>	<p><b>15L</b></p>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) Certified Ethical Hacker Study Guide v9, Sean-Philip Oriyano, Sybex; Study Guide Edition,2016</li> <li>2) CEH official Certified Ethical Hacking Review Guide, Wiley India Edition, 2007</li> </ol> <p><b>Additional Reference(s):</b></p>		

- 1) Certified Ethical Hacker: Michael Gregg, Pearson Education, 1<sup>st</sup> Edition, 2013
- 2) Certified Ethical Hacker: Matt Walker, TMH, 2011
- 3) [http://www.pentest-standard.org/index.php/PTES\\_Technical\\_Guidelines](http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines)
- 4) [https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_2017\\_Project](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project)
- 5) [https://www.owasp.org/index.php/Mobile\\_Top\\_10\\_2016-Top\\_10](https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10)
- 6) [https://www.owasp.org/index.php/OWASP\\_Testing\\_Guide\\_v4\\_Table\\_of\\_Contents](https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents)
- 7) [https://www.owasp.org/index.php/OWASP\\_Secure\\_Coding\\_Practices\\_-\\_Quick\\_Reference\\_Guide](https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide)
- 8) <https://cve.mitre.org/>
- 9) <https://access.redhat.com/blogs/766093/posts/2914051>
- 10) <http://resources.infosecinstitute.com/applications-threat-modeling/#gref>
- 11) <http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html>

## Suggested List of Practical – SEMESTER VI

<b>Course:</b> <b>USCSP601</b>	<b>(Credits : 02 Lectures/Week:06)</b> <b>Practical of Elective-I</b>	
<b>USCS601: Wireless Sensor Networks and Mobile Communication</b>		
<p><i>Practical experiments require software tools like INET Framework for OMNeT++, NetSim , TOSSIM, Cisco packet tracer 6.0 and higher version.</i></p> <ol style="list-style-type: none"> <li>1. Understanding the Sensor Node Hardware. (For Eg. Sensors, Nodes(Sensor mote), Base Station, Graphical User Interface.)</li> <li>2. Exploring and understanding TinyOS computational concepts:- Events, Commands and Task.             <ul style="list-style-type: none"> <li>- nesC model</li> <li>- nesC Components</li> </ul> </li> <li>3. Understanding TOSSIM for             <ul style="list-style-type: none"> <li>- Mote-mote radio communication</li> <li>- Mote-PC serial communication</li> </ul> </li> <li>4. Create and simulate a simple adhoc network</li> <li>5. Understanding, Reading and Analyzing Routing Table of a network.</li> <li>6. Create a basic MANET implementation simulation for Packet animation and Packet Trace.</li> <li>7. Implement a Wireless sensor network simulation.</li> <li>8. Create MAC protocol simulation implementation for wireless sensor Network.</li> <li>9. Simulate Mobile Adhoc Network with Directional Antenna</li> <li>10. Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them.</li> </ol>		
<b>USCS602: Cloud Computing</b>		
<ol style="list-style-type: none"> <li>1. Study and implementation of Infrastructure as a Service.</li> <li>2. Installation and Configuration of virtualization using KVM.</li> <li>3. Study and implementation of Infrastructure as a Service</li> <li>4. Study and implementation of Storage as a Service</li> <li>5. Study and implementation of identity management</li> <li>6. Study Cloud Security management</li> </ol>		

7. Write a program for web feed.
8. Study and implementation of Single-Sign-On.
9. User Management in Cloud.
10. Case study on Amazon EC2/Microsoft Azure/Google Cloud Platform

### **USCS603: Cyber Forensics**

1. Creating a Forensic Image using FTK Imager/Encase Imager :
  - Creating Forensic Image
  - Check Integrity of Data
  - Analyze Forensic Image
2. Data Acquisition:
  - Perform data acquisition using:
    - USB Write Blocker + Encase Imager
    - SATA Write Blocker + Encase Imager
    - Falcon Imaging Device
3. Forensics Case Study:
  - Solve the Case study (image file) provide in lab using Encase Investigator or Autopsy
4. Capturing and analyzing network packets using Wireshark (Fundamentals) :
  - Identification the live network
  - Capture Packets
  - Analyze the captured packets
5. Analyze the packets provided in lab and solve the questions using Wireshark :
  - What web server software is used by www.snopes.com?
  - About what cell phone problem is the client concerned?
  - According to Zillow, what instrument will Ryan learn to play?
  - How many web servers are running Apache?
  - What hosts (IP addresses) think that jokes are more entertaining when they are explained?
6. Using Sysinternals tools for Network Tracking and Process Monitoring :
  - Check Sysinternals tools



- Monitor Live Processes
- Capture RAM
- Capture TCP/UDP packets
- Monitor Hard Disk
- Monitor Virtual Memory
- Monitor Cache Memory
- 7. Recovering and Inspecting deleted files
  - Check for Deleted Files
  - Recover the Deleted Files
  - Analyzing and Inspecting the recovered files

Perform this using recovery option in ENCASE and also Perform manually through command line
- 8. Acquisition of Cell phones and Mobile devices
- 9. Email Forensics
  - Mail Service Providers
  - Email protocols
  - Recovering emails
  - Analyzing email header
- 10. Web Browser Forensics
  - Web Browser working
  - Forensics activities on browser
  - Cache / Cookies analysis
  - Last Internet activity

<b>Course:</b>	<b>(Credits : 02 Lectures/Week:06)</b>	
<b>USCSP602</b>	<b>Practical of Elective-II</b>	
<b>USCS604: Information Retrieval</b>		
<i>Practical may be done using software/tools like Python / Java / Hadoop</i>		
<ol style="list-style-type: none"> <li>1. Write a program to demonstrate bitwise operation.</li> <li>2. Implement Page Rank Algorithm.</li> <li>3. Implement Dynamic programming algorithm for computing the edit distance between</li> </ol>		

strings s1 and s2. (Hint. Levenshtein Distance)

4. Write a program to Compute Similarity between two text documents.
5. Write a map-reduce program to count the number of occurrences of each alphabetic character in the given dataset. The count for each letter should be case-insensitive (i.e., include both upper-case and lower-case versions of the letter; Ignore non-alphabetic characters).
6. Implement a basic IR system using Lucene.
7. Write a program for Pre-processing of a Text Document: stop word removal.
8. Write a program for mining Twitter to identify tweets for a specific period and identify trends and named entities.
9. Write a program to implement simple web crawler.
10. Write a program to parse XML text, generate Web graph and compute topic specific page rank.

### **USCS605: Digital Image Processing**

*Practical need to be performed using Scilab under Linux or Windows*

1. 2D Linear Convolution, Circular Convolution between two 2D matrices
2. Circular Convolution expressed as linear convolution plus alias
3. Linear Cross correlation of a 2D matrix, Circular correlation between two signals and Linear auto correlation of a 2D matrix, Linear Cross correlation of a 2D matrix
4. DFT of 4x4 gray scale image
5. Compute discrete cosine transform, Program to perform KL transform for the given 2D matrix
6. Brightness enhancement of an image, Contrast Manipulation, image negative
7. Perform threshold operation, perform gray level slicing without background
8. Image Segmentation
9. Image Compression
10. Binary Image Processing and Colour Image processing

### **USCS606:Data Science**

*Practical shall be performed using R*

1. Practical of Data collection, Data curation and management for Unstructured data (NoSQL)

2. Practical of Data collection, Data curation and management for Large-scale Data system (such as MongoDB)
3. Practical of Principal Component Analysis
4. Practical of Clustering
5. Practical of Time-series forecasting
6. Practical of Simple/Multiple Linear Regression
7. Practical of Logistics Regression
8. Practical of Hypothesis testing
9. Practical of Analysis of Variance
10. Practical of Decision Tree

**Course:**  
**USCSP603**

**(Credits : 01 Lectures/Week: 03)**  
**Project Implementation**

*Please Refer to Project Implementation Guidelines*

**Course:**  
**USCSP604**

**(Credits : 01 Lectures/Week: 03)**  
**Practical of Skill Enhancement**

**USCS607 : Ethical Hacking**

1. Use Google and Whois for Reconnaissance
2. a) Use CrypTool to encrypt and decrypt passwords using RC4 algorithm  
b) Use Cain and Abel for cracking Windows account password using Dictionary attack and to decode wireless network passwords
3. a) Run and analyze the output of following commands in Linux – ifconfig, ping, netstat, traceroute  
b) Perform ARP Poisoning in Windows
4. Use NMap scanner to perform port scanning of various forms – ACK, SYN, FIN, NULL, XMAS
5. a) Use Wireshark (Sniffer) to capture network traffic and analyze  
b) Use Nemesy to launch DoS attack
6. Simulate persistent cross-site scripting attack
7. Session impersonation using Firefox and Tamper Data add-on

8. Perform SQL injection attack
9. Create a simple keylogger using python
10. Using Metasploit to exploit (Kali Linux)

## Project Implementation Guidelines

1. A learner is expected to carry out two different projects: one in Semester V and another in Semester VI.
2. A learner can choose any topic which is covered in Semester I- semester VI or any other topic with the prior approval from head of the department/ project in charge.
3. The Project has to be performed individually.
4. A learner is expected to devote around three months of efforts in the project.
5. The project can be application oriented/web-based/database/research based.
6. It has to be an implemented work; just theoretical study will not be acceptable.
7. A learner can choose any programming language, computational techniques and tools which have been covered during BSc course or any other with the prior permission of head of the department/ project guide.
8. A project guide should be assigned to a learner. He/she will assign a schedule for the project and hand it over to a learner. The guide should oversee the project progress on a weekly basis by considering the workload of 3 lectures as assigned.
9. The quality of the project will be evaluated based on the novelty of the topic, scope of the work, relevance to the computer science, adoption of emerging techniques/technologies and its real-world application.
10. A learner has to maintain a project report with the following subsections
  - a) Title Page
  - b) Certificate

A certificate should contain the following information –

- The fact that the student has successfully completed the project as per the syllabus and that it forms a part of the requirements for completing the BSc degree in computer science of University of Mumbai.
- The name of the student and the project guide
- The academic year in which the project is done
- Date of submission,
- Signature of the project guide and the head of the department with date along with the department stamp,

- Space for signature of the university examiner and date on which the project is evaluated.
- c) Self-attested copy of Plagiarism Report from any open source tool.
- d) Index Page detailing description of the following with their subsections:
- Title: A suitable title giving the idea about what work is proposed.
  - Introduction: An introduction to the topic giving proper back ground of the topic.
  - Requirement Specification: Specify Software/hardware/data requirements.
  - System Design details : Methodology/Architecture/UML/DFD/Algorithms/protocols etc. used(whichever is applicable)
  - System Implementation: Code implementation
  - Results: Test Cases/Tables/Figures/Graphs/Screen shots/Reports etc.
  - Conclusion and Future Scope: Specify the Final conclusion and future scope
  - References: Books, web links, research articles, etc.
11. The size of the project report shall be around twenty to twenty five pages, excluding the code.
12. The Project report should be submitted in a spiral bound form
13. The Project should be certified by the concerned Project guide and Head of the department.
14. A learner has to make a presentation of working project and will be evaluated as per the Project evaluation scheme

# Scheme of Examination

## 1. Theory:

### I. Internal 25 Marks :

#### a) Test – 20 Marks

20 marks Test – Duration 40 mins

It will be conducted either using any open source learning management system like Moodle (Modular object-oriented dynamic learning environment)

**OR**

A test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.

- b) **5 Marks** – Active participation in routine class instructional deliveries  
Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

### II. External 75 Marks as per University Guidelines

#### 11. Practical and Project Examination:

There will be separate Practical examination for Elective-I, II, Skill enhancement and project of these Elective-I 100, Elective-II: 100 and Skill Enhancement: 50 and Project Implementation: 50.

In the Practical Examination of Elective-I and II, the student has to perform practical on each of the subjects chosen. The Marking Scheme for each of the Elective is given below:

	<b>Subject Code</b>	<b>Experiment-I</b>	<b>Experiment-II</b>	<b>Total Marks</b>
<b>Elective-I</b>	<b>USCSP501/ USCSP601</b>	Experiment-40+Journal-5 +viva-5 Total:50M	Experiment-40+Journal-5+viva-5 5 Total:50M	100 M
<b>Elective-II</b>	<b>USCSP502/ USCSP602</b>	Experiment-40+Journal-5 +viva-5 Total:50M	Experiment-40+Journal-5+viva-5 5 Total:50M	100 M

<b>Project Implementation</b>	<b>USCSP503/ USCSP603</b>	<b>**Project Evaluation Scheme</b>	<b>50M</b>
<b>Skill Enhancement</b>	<b>USCSP504/ USCSP604</b>	Experiment-40+Journal:5+viva-5 Total-50M	<b>50M</b>
<b>Total Marks</b>			<b>300M</b>

**(Certified Journal is compulsory for appearing at the time of Practical Examination)**

**\*\*Project Evaluation Scheme:**

<b>Presentation</b>	<b>Working of the Project</b>	<b>Quality of the Project</b>	<b>Viva</b>	<b>Documentation</b>
10Marks	10 Marks	10 Marks	10 Marks	10Marks

**(Certified Project Document is compulsory for appearing at the time of Project Presentation)**

\*\*\*\*\*



**CIRCULAR:-**

The Head of the University Departments of Computer Science , the Principals of the affiliated Colleges in Science and the Directors of recognized Science Institutions concerned are hereby informed that in continuation syllabi relating to Bachelor of Science degree Course passed by the Academic Council at its meeting held on 27/2/2013, vide item No. 4.50 and recommendations made by the Ad - hoc Board of Studies in Computer Science at its meeting held on 5/5/2017 has been accepted by the Academic Council at its meeting held on 11<sup>th</sup> May, 2017, vide item no. 4.211 and that in accordance therewith, the revised syllabus as per the (CBCS) for (Sem V & VI) of B.Sc programme in the Course of Computer Science, which is available on the University's website ([www.mu.ac.in](http://www.mu.ac.in)) and that the same has been brought into force with effect from the academic year 2017-18.

MUMBAI - 400 032

27<sup>th</sup> July, 2017

*Shree*  
25/7/17  
REGISTRAR

To,

The Head of the University Departments of Computer Science, the Principals of the affiliated Colleges in Science and the Directors of Recognized Institutions concerned.

A.C/4.211/11.05.2017

No. UG/108-A of 2017

MUMBAI-400 032

27<sup>th</sup> July, 2017

Copy forwarded with compliments for information to :-

- 1) The Co-ordinator, Faculty of Science,
- 2) The Offg. Director, Board of Examinations and Evaluation,
- 3) The Director of Board of Studies Development,
- 4) The Professor-cum-Director, Institute of Distance and Open Learning.
- 5) The Co-Ordinator, University Computerization Centre.

*Shree*  
25/7/17  
REGISTRAR

....PTO

**UNIVERSITY OF MUMBAI**



**Syllabus for Sem V & VI**  
**Program: B.Sc.**  
**Course: Computer Science**

(Credit Based Semester and Grading System with  
effect from the academic year 2017-2018 )

## **Preamble**

In this era of Computerisation, Digitalization and Automation, there is barely any field of research or any industry left that is not benefitting from Computer Science or Information Technology. The Graduation course in Computer Science holds big importance in cultivating skilled professionals. The courses of third-year of B.Sc. (Computer Science) are therefore designed in a such way which will develop the students not only as a professional developer but also with the view of research oriented.

To enhance programming skills among students Programming holds key indispensable position in any curriculum of Computer Science. It is essential for the learners to know how to use Object Oriented paradigm. This is covered during course of Advanced Java in both fifth and sixth semesters. There is also one dedicated course for Mobile Development catering to modern day needs of Mobile platforms and applications.

Today's world is about connectivity and shared computing. A course in Data Communications and Networking is therefore very apt for the students who are gearing for professional world of applications. Along with these courses Web Computing courses gives enough idea about theories and fundamentals of building robust web interfaces.

**T.Y.B.Sc. (Semester V and VI)**  
**Computer Science Syllabus**  
**(Credit Based Semester and Grading System)**  
**To be implemented from academic year 2017-2018**

<b>SEMESTER V</b>			
<b>Course</b>	<b>Topics</b>	<b>Credits</b>	<b>L / Week</b>
USCS501	Data Communication and Networking	2.5	4
USCS502	Advanced Java Programming– I	2.5	4
USCS503	Mobile Application Development	2.5	4
USCS504	Data Management using PL/SQL-I	2.5	4
USCSP501	Practical of USCS501 + USCS502	3	8
USCSP502	Practical of USCS503 + USCS504	3	8

<b>SEMESTER VI</b>			
<b>Course</b>	<b>Topics</b>	<b>Credits</b>	<b>L / Week</b>
USCS601	Advanced Networking & Security	2.5	4
USCS602	Advanced Java Programming – II	2.5	4
USCS603	Software Engineering and Testing	2.5	4
USCS604	Data Management using PL/SQL-II	2.5	4
USCSP601	Practical of USCS602 + USCS604	3	8
USCSP602	Practical of USCS601 + USCS603	3	8

**SEMESTER V**  
**THEORY**

<b>Course:</b> <b>USCS501</b>	<b>TOPICS (Credits : 2.5 Lectures/Week: 04)</b> <b>Data Communication and Networking</b>	
<b>Unit I</b>	<p><b>Introduction</b> - Data Communication, Networks, Internet, Intranet, Protocols, OSI &amp; TCP/IP Models, Addressing</p> <p><b>Physical Layer</b> - Signals, Analog, Digital, Analog VS Digital, Transmission Impairment, Data Rate Limits, Performance</p> <p><b>Digital Transmission</b> - Line Coding (Unipolar, Polar, Biphasic), Block Coding(4B/5B Encoding), Analog to digital conversion, PCM, Transmission Modes,</p> <p><b>Analog Transmission</b> - Digital to analog conversion(ASK,FSK,PSK, QAM), Analog to Analog conversion</p>	<b>15L</b>
<b>Unit II</b>	<p><b>Multiplexing</b> - FDM, WDM, Synchronous TDM(time slots &amp; frames, interleaving, data rate management),</p> <p><b>Spread Spectrum</b> - FHSS, DSSS</p> <p><b>Transmission Media</b> - Guided &amp; Unguided</p> <p><b>Switching</b> - Switching, Circuit-Switched Networks, Datagram networks, Concept of Virtual circuit networks, structure of circuit switch &amp; packet switch, Concepts of DSL &amp; ADSL</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Data Link Layer</b> -Error correction &amp; detection, Types of errors, Detection VS Correction, Block Coding, Hamming Distance, Linear Block codes(single parity check, hamming codes), Cyclic codes, CRC Encoder &amp; Decoder, CRC Polynomial &amp; its degree, Checksum</p> <p><b>Data Link Control &amp; Protocols</b> - Framing, Flow &amp; Error Control, Simplex, Stop-N-Wait, Stop-N-Wait ARQ, Go Back N ARQ, Selective Repeat ARQ, Piggybacking</p> <p>HDLC &amp; PPP- HDLC Modes, HDLC Frames, PPP, PPP Transition states</p>	<b>15L</b>
<b>Unit IV</b>	<p><b>Multiple Access</b> - Random(CSMA), Controlled(Reservation, Polling,</p>	<b>15L</b>

	<p>Token Passing), Channelization(FDMA, TDMA, CDMA)</p> <p><b>Wired LAN</b> - LLC, MAC, Ethernet, Ethernet frame, Addressing, Concept of MBaseN Ethernet, Bridged, Switched, Full Duplex Ethernet, Concept of Fast &amp; Gigabit Ethernet</p> <p><b>Wireless LAN</b> - Introduction to WLAN(Architecture, Hidden, Exposed Station Problem), Introduction to Bluetooth &amp; Architecture, Cellular telephony, Concept of 1G, 2G, 3G cellular telephony</p> <p><b>Connecting Devices</b> - Repeaters, Hubs, Bridges, Spanning tree algorithm, Two &amp; Three layer Switches, Routers, Gateways, Backbone networks, Concept of VLAN</p>	
<p><b>Text-book(s):</b></p> <ol style="list-style-type: none"> <li>1) Data Communication &amp; Networking (Forouzan), Tata McGraw-Hill Education</li> <li>2) Computer Networks - Andrew Tanenbaum, PHI</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Computer Network, Bhushan Trivedi, Oxford University Press</li> <li>2) Computer Networks and Internets - Douglas Comer, Prentice Hall</li> <li>3) Computer Networking, Kurose, Ross, Pearson</li> </ol>		

<b>Course:</b> USCS502	<b>TOPICS (Credits : 2.5 Lectures/Week: 04)</b> <b>Advanced Java Programming– I</b>	
<b>Unit I</b>	<b>Swing Components – I:</b> Introduction to JFC and Swing, Features of the Java Foundation Classes, Swing API Components, JComponent Class, Windows, Dialog Boxes, and Panels, Labels, Buttons, Check Boxes, Menus, Pane, JScrollPane, Desktop pane, Scrollbars, Lists and Combo Boxes, Text-Entry Components.	<b>15L</b>
<b>Unit II</b>	<b>Swing Components – II:</b> Toolbars, Implementing Action interface, Colors and File Choosers, Tables and Trees, Printing with 2D API and Java Print Service API. Schedules Tasks using JVM, Thread-safe variables, Communication between threads. <b>Event Handling:</b> The Delegation Event Model, Event classes (ActionEvent, FocusEvent, InputEvent, ItemEvent, KeyEvent,	<b>15L</b>

	MouseEvent, MouseWheelEvent, TextEvent, WindowEvent) and various listener interfaces (ActionListener, FocusListener, ItemListener, KeyListener, MouseListener, MouseMotionListener, MouseWheelListener, TextListener, WindowFocusListener, WindowListener)	
<b>Unit III</b>	<b>JDBC:</b> JDBC Introduction, JDBC Architecture, Types of JDBC Drivers, The Connectivity Model, The java.sql package, Navigating the ResultSet object's contents, Manipulating records of a ResultSet object through User Interface , The JDBC Exception classes, Database Connectivity, Data Manipulation (using Prepared Statements, Joins, Transactions, Stored Procedures), Data navigation.	<b>15L</b>
<b>Unit IV</b>	<b>Networking with JAVA:</b> Overview of Networking, Working with URL, Connecting to a Server, Implementing Servers, Serving multiple Clients, Sending E-Mail, Socket Programming, Internet Addresses, URL Connections. Accessing Network interface parameters, Posting Form Data, Cookies, Overview of Understanding the Sockets Direct Protocol.  Introduction to distributed object system, Distributed Object Technologies, RMI for distributed computing, RMI Architecture, RMI Registry Service, Parameter Passing in Remote Methods, Creating RMI application, Steps involved in running the RMI application, Using RMI with Applets.	<b>15L</b>
<b>Text book(s):</b>		
<ol style="list-style-type: none"> <li>1) Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD)</li> <li>2) Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR</li> <li>3) Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill</li> </ol>		
<b>Additional Reference(s):</b>		
<ol style="list-style-type: none"> <li>1) The Java Tutorials of Sun Microsystems Inc.</li> </ol>		

<b>Course:</b> <b>USCS503</b>	<b>TOPICS (Credits : 2.5 Lectures/Week:04)</b> <b>Mobile Application Development</b>	
<b>Unit I</b>	<p><b>Introduction to Mobile Application Development</b></p> <p><b>Introduction to Mobile Computing</b> - Definition and general overview of Mobile and Cell Phone Technologies - CDMA, GSM, 3G, 4G, Types of mobile computing devices - PDA, Pagers, Mobiles, etc.</p> <p><b>History of mobile platforms</b> - J2ME, BB, Android, Windows Mobile, Windows Phone, etc.</p> <p><b>The Android Platform:</b> Introduction to the Android platform, Architecture, Android components, Development Tools – SDK, ADB, Gradle, etc. Installing Android Studio IDE, and developing first app</p> <p><b>Activities and Lifecycle, Fragments and Intents</b> - Working with Activities-creating activity, starting activity, managing life cycle of activity, applying themes and styles, displaying dialog in activity; Using Intents-exploring intent objects, resolution, filters passing data using objects in intents; Fragments, Intent Object to Invoke Built-in Application</p>	<b>15L</b>
<b>Unit II</b>	<p><b>UI Design:</b> Display Orientation, Views and ViewGroups, Layouts, Action Bars and Navigation Drawers, Android Layout Managers - LinearLayout, RelativeLayout, ScrollView, TableLayout, FrameLayout, ActionBar, Working with Views- TextView, EditText View, Button View, RadioButton View, CheckBox View, ImageButton View, ToggleButton View, RatingBar View</p> <p><b>UI Events:</b> Understanding Android Events, Using the android:onClick Resource, Event Listeners and Callback Methods, Event Handling, The Event Listener and Callback Method, Intercepting Touch Events, Implementing Common Gesture Detection</p> <p><b>Data binding in applications</b> - Introduction to data binding in Android, What is an Adapter?, Adapter Views - ListView Class, Spinner, Gallery View, AutoTextCompleteView, GridView</p> <p><b>Displaying Pictures and Menus with Views</b> - Working with Image Views, Designing Context Menu for Image View, Embedding Web Browser in an</p>	<b>15L</b>



	<p>Activity using WebView, Notifying the User</p> <p><b>Data Persistence</b> - The Data Storage Options, Internal Storage, External Storage, Using the SQLite Database - CRUD, Working with Content Providers</p>	
<b>Unit III</b>	<p><b>Networking in Android:</b> Accessing the network, Permission to access the network, Checking Network Availability, Sending Email, consuming web services using HTTP</p> <p><b>Location-Based Services</b> - Displaying Maps, Getting Location Data, monitoring a Location, Google Maps API, Using the Geocoder.</p> <p><b>Using Multimedia</b> — Audio, Video, and the Camera</p> <p>Playing audio and video, recording audio and video, Using Camera for Taking Pictures, Using Media Player</p> <p><b>Telephony and SMS:</b> Handling Telephony, Handling SMS, Sending SMS Using Intent</p>	<b>15L</b>
<b>Unit IV</b>	<p><b>Working with Bluetooth and Wi-Fi</b> - BluetoothAdapter and Managing Wi-Fi connectivity using WifiManager</p> <p><b>Threads and Thread Handlers</b> - Introduction to Threads, Worker threads - AsyncTask, interprocess communication and Services</p> <p><b>Working with Graphics and Animation:</b> Working with Graphics, Using the Drawable Object, Using the ShapeDrawable Object, Concept of Hardware Acceleration, Working with Animations</p> <p><b>Advanced Development</b> - Cloud to Device Messaging using Google Firebase Cloud Messaging, Publishing the App, Best Practices for Performance</p>	<b>15L</b>
<p><b>Text book(s):</b></p> <ol style="list-style-type: none"> <li>1) Professional Android™ 4 Application Development, Reto Meier, John Wiley &amp; Sons, Inc. 2012.</li> <li>2) Android Application Development, Black Book, Pradeep Kothari, Kogent Learning Solutions, DreamTech Press</li> <li>3) Google Android Developers - <a href="https://developer.android.com/index.html">https://developer.android.com/index.html</a></li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Expert Android Studio, Murat Yenar, Onur Dunder, Wrox</li> </ol>		

- 2) Android Studio Cookbook, Mike van Drongelen, PACKT Publication
- 3) Android Programming for Beginners by John Horton (Author), PACKT Publication
- 4) Hello, Android: Introducing Google's Mobile Development Platform, Third Edition, Ed

Course: USCS504	TOPICS (Credits : 2.5 Lectures/Week: 04) Data Management using PL/SQL-I	
<b>Unit I</b>	<p><b>Fundamentals of PL SQL</b> Introduction to SQL Developer, Introduction to PL/SQL, PL/SQL Overview, Benefits of PL/SQL, Subprograms, Overview of the Types of PL/SQL blocks, Create a Simple Anonymous Block, Generate Output from a PL/SQL Block</p> <p><b>SQL Identifiers</b> List the different Types of Identifiers in a PL/SQL subprogram, Usage of the Declarative Section to define Identifiers, Use variables to store data, Identify Scalar Data Types, The %TYPE Attribute, Bind Variables, Sequences in PL/SQL Expressions</p> <p><b>Write Executable Statements</b> Describe Basic PL/SQL Block Syntax Guidelines, Comment Code, Deployment of SQL Functions in PL/SQL, Nested Blocks, Identify the Operators in PL/SQL.</p>	<b>15L</b>
<b>Unit II</b>	<p><b>Conversion Functions:</b> implicit and explicit data type conversion, Describe the TO_CHAR, TO_NUMBER, and TO_DATE conversion functions, Nesting multiple functions</p> <p><b>Control Structures:</b> Conditional processing Using IF Statements, Conditional processing Using CASE Statements, Use simple Loop Statement, Use While Loop Statement, Use For Loop Statement, Describe the Continue Statement</p> <p><b>Composite Data Types</b> Use PL/SQL Records, The %ROWTYPE Attribute, Insert and Update with PL/SQL Records, Associative Arrays (INDEX BY Tables), Examine INDEX BY Table Methods, Use INDEX BY Table of Records</p>	<b>15L</b>

<p><b>Unit III</b></p>	<p><b>Exception Handling</b> Understand Exceptions, Handle Exceptions with PL/SQL, Trap Predefined Oracle Server Errors, Trap Non-Predefined Oracle Server Errors, Trap User-Defined Exceptions, Propagate Exceptions, RAISE_APPLICATION_ERROR Procedure</p> <p><b>Stored Procedures and Functions</b> Understand Stored Procedures and Functions, Differentiate between anonymous blocks and subprograms, Create a Simple Procedures, Create a Simple Procedure with IN parameter, Create a Simple Function, Execute a Simple Procedure, Execute a Simple Function.</p> <p><b>Invoke SELECT Statements in PL/SQL to Retrieve data:</b> Data Manipulation in the Server Using PL/SQL, SQL Cursor concept, Usage of SQL Cursor Attributes to Obtain Feedback on DML, Save and Discard Transactions.</p>	<p><b>15L</b></p>
<p><b>Unit IV</b></p>	<p><b>Explicit Cursors</b> What are Explicit Cursors?, Declare the Cursor, Open the Cursor, Fetch data from the Cursor, Close the Cursor, Cursor FOR loop, Explicit Cursor Attributes, FOR UPDATE Clause and WHERE CURRENT Clause</p> <p><b>Collections</b> Index-by tables or Associative array, Nested table, Variable-size array or Varray Strings, Date and Time functions, arrays</p>	<p><b>15L</b></p>
<p><b>Text book(s):</b></p> <ol style="list-style-type: none"> <li>1) Oracle SQL and PL/SQL, Joel Murach</li> <li>2) PL/SQL Language Reference 11g, Sheila Moore, E. Belden,</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Ivan Bayross, "SQL,PL/SQL -The Programming language of Oracle", B.P.B. Publications</li> <li>2) Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner’s Guide, Tata McGraw-Hill.</li> <li>3) Martin Gruber, "Understanding SQL", B.P.B. Publications.</li> <li>4) George Koch and Kevin Loney ,ORACLE "The Complete Reference", Tata McGraw Hill,New Delhi</li> <li>5) <a href="https://docs.oracle.com">https://docs.oracle.com</a></li> </ol>		

## Suggested List of Practical – SEMESTER V

<b>Course:</b> <b>USCSP501</b>	<b>(Credits : 03 Practical/Week: 08)</b> <b>USCS501+ USCS502</b>	
<b>Data Communication and Networking</b>		
<ol style="list-style-type: none"><li>1. Study of URL, InetAddress and its members</li><li>2. Study of URLConnection &amp; to read the contents.</li><li>3. Study of URLConnection &amp; to write to it.</li><li>4. Study of Connection-less approach using datagram-approach</li><li>5. Study of connection-oriented approach using ServerSocket</li><li>6. Creating server process using ServerSocket</li><li>7. Sending Email through Java</li><li>8. Designing RMI Application</li></ol>		
<b>Advanced JAVA Programming-I</b>		
<ol style="list-style-type: none"><li>1. Using Basic Swing Controls</li><li>2. Using JScrollPane, JTabbedPane, JDesktopPane</li><li>3. Using Common Dialog Boxes</li><li>4. Using JTable and JTree</li><li>5. Creating Table in database</li><li>6. Inserting data in tables &amp; Displaying data</li><li>7. Using ResultSetMetaData</li><li>8. Using Prepared Statements</li></ol>		

<b>Course:</b> <b>USCSP502</b>	<b>(Credits : 03 Practical/Week: 08)</b> <b>USCS503+ USCS504</b>	
<b>Mobile Application Development</b>		
<ol style="list-style-type: none"> <li>1. Design an application representing a simple calculator.</li> <li>2. Develop an application for working with Menus and Screen Navigation</li> <li>3. Develop an application for working with Notifications</li> <li>4. Develop an application demonstrating Internal Storage to store private data on the device memory.</li> <li>5. Design a simple to-do list application using SQLite</li> <li>6. Develop an application for connecting to the internet and sending email.</li> <li>7. Develop an application for working with graphics and animation.</li> <li>8. Develop an application for working with device camera.</li> <li>9. Develop an application for working with location based services.</li> <li>10. Using Worker thread write Android code for a click listener that downloads an image from a separate thread and displays it in an ImageView.</li> </ol>		
<b>Data Management using PL/SQL-I</b>		
<ol style="list-style-type: none"> <li>1. Writing Anonymous PL/SQL Block with basic programming construct by including following: <ol style="list-style-type: none"> <li>a. Sequential Statements</li> <li>b. unconstrained loop</li> </ol> </li> <li>2. Writing PL/SQL Blocks with basic programming constructs by including following: <ol style="list-style-type: none"> <li>a. CONSTANT</li> <li>b. NOT NULL</li> <li>c. DEFAULT</li> <li>d. %TYPE and % ROWTYPE Attribute.</li> </ol> </li> <li>3. Writing PL/SQL Blocks with basic programming constructs by including following conversion functions: TO_CHAR, TO_NUMBER, and TO_DATE, blocks on strings, date and time functions, and arrays.</li> <li>4. Writing PL/SQL Blocks with basic programming constructs by including following: <ol style="list-style-type: none"> <li>a. If...then...Else, IF...ELSIF...ELSE... END IF</li> <li>b. Case statement</li> </ol> </li> </ol>		

5. Writing PL/SQL Blocks with basic programming constructs for following Iterative Structure:
  - a. While-loop Statements
  - b. For-loop Statements.
6. Writing Exception Handling with PL/SQL.
  - a. Exception Types (implicitly raised, Explicitly raised)
  - b. Trapping Exceptions (WHEN exception1, WHEN OTHERS)
  - c. Predefined Exception
    - NO\_DATA\_FOUND
    - TOO\_MANY\_ROWS
    - INVALID\_CURSOR
    - ZERO\_DIVIDE
    - DUP\_VAL\_ON\_INDEX
7. Writing Procedures in PL/SQL Block (IN, OUT, INOUT, DEFAULT keywords).
  - a. Create an empty procedure, replace a procedure and call procedure
  - b. Create a stored procedure and call it
  - c. Define procedure to insert data
  - d. A forward declaration of procedure
8. Writing Functions in PL/SQL Block.
  - a. Define and call a function
  - b. Define and use function in select clause,
  - c. Call function in dbms\_output.put\_line
  - d. Recursive function
  - e. Count Employee from a function and return value back
  - f. Call function and store the return value to a variable
9. Writing PL/SQL Block for
  - a. Declare and use Association Array
  - b. Varray
  - c. Nested Tables
10. Writing PL/SQL Block for Cursors
  - a. Cursor attributes:%ROWCOUNT,%FOUND,%NOTFOUND,%ISOPEN
  - b. Cursor with sub queries
  - c. Combination of PL/SQL, cursor and for loop
  - d. Parameterized cursors, Cursor Variables

## SEMESTER VI

### THEORY

<b>Course:</b> USCS601	<b>TOPICS (Credits : 2.5 Lectures/Week: 04)</b> <b>Advanced Networking &amp; Security</b>	
<b>Unit I</b>	<p><b>Network Layer</b> -Logical addressing, IPv4 Addresses, Classful &amp; Classless addresses, NAT, IPv6 Addressing,</p> <p><b>Network layer protocol</b> - Internetworking, IPv4, IPv4 protocol packet format, IPv6 Protocol &amp; Packet format, IPv4 VS IPv6, Transition from IPv4 to IPv6, Address Resolution protocols(ARP, RARP), BOOTP, DHCP, <b>Routing Protocols</b> - Delivery, forwarding, routing, types of routing, routing tables, Unicast Routing, Unicast Routing protocols, RIP, Concepts of OSPF, BGP &amp; Multicast Routing</p>	<b>15L</b>
<b>Unit II</b>	<p><b>Transport Layer</b> - Process to process delivery, UDP, TCP <b>Congestion Control &amp; Quality of Service</b>- Data traffic, Congestion, Congestion Control(Open Loop, Closed Loop &amp; Congestion control in TCP), QoS and Flow Characteristics</p> <p><b>Application Layer</b> - DNS, Remote Logging(Telnet), SMTP, FTP, WWW, HTTP</p>	<b>15L</b>
<b>Unit III</b>	<p>System and network security: Introduction to system and network security, security attacks, security services and mechanisms.</p> <p><b>Malicious software and Internet Security:</b> viruses and related threats, virus countermeasures, denial of service attacks, <i>Hacking</i>, Security policies and plan, Strategies for a secure network.</p> <p><b>Firewall and Intrusion Detection:</b> Firewalls and their types, DMZ, Limitations of firewalls, Intruders, Intrusion detection (Host based, Networked, Distributed), IDS.</p>	<b>15L</b>
<b>Unit IV</b>	<p><b>Cryptography:</b> Traditional and Modern Symmetric-Key Ciphers, DES and AES, Asymmetric -Key Cryptography, RSA and ELGAMAL cryptosystems. Message Digest, Digital Signature, Key Management</p> <p><b>Network Security:</b> Security at Application Layer (E-MAIL, PGP and S/MIME), Security at Transport Layer (SSL and TLS), Security at</p>	<b>15L</b>

	Network Layer (IPSec).	
<b>Text book(s):</b>		
<ol style="list-style-type: none"> <li>1) Data Communication &amp; Networking (Forouzan) , Tata McGraw-Hill Education</li> <li>2) Cryptography &amp; Network Security, Behrouz A. Forouzan, Tata McGraw-Hill,</li> <li>3) Network security essentials-applications and standards, William Stallings, Third Edition, Pearson Education</li> </ol>		
<b>Additional Reference(s):</b>		
<ol style="list-style-type: none"> <li>1) Computer Networks and Internets - Douglas Comer, Prentice Hall</li> <li>2) Computer Networks - Andrew Tanenbaum, Prentice Hall</li> <li>3) Computer Network, Bhushan Trivedi, Oxford University Press</li> </ol>		

<b>Course:</b> USCS602	<b>TOPICS (Credits : 2.5 Lectures/Week: 04)</b> <b>Advanced Java Programming-II</b>	
<b>Unit I</b>	<b>Servlet:</b> What Is a Servlet? The Example Servlets, Servlet Life Cycle, Sharing Information, Initializing a Servlet, Writing Service Methods, Filtering Requests and Responses, Invoking Other Web Resources, Accessing the Web Context, Maintaining Client State, Finalizing a Servlet.	<b>15L</b>
<b>Unit II</b>	<b>JSP:</b> What Is a JSP Page?, The Example JSP Pages, The Life Cycle of a JSP Page, Creating Static Content, Creating Dynamic Content, Unified Expression Language, JavaBeans Components, JavaBeans Concepts, Using NetBeans GUI Builder Writing a Simple Bean, Properties: Simple Properties, Using Custom tags, Reusing content in JSP Pages, Transferring Control to Another Web Component, Including an Applet.	<b>15L</b>
<b>Unit III</b>	<b>EJB:</b> Introduction to EJB, Benefits of EJB, Types of EJB, Session Bean: State Management Modes; Message-Driven Bean, Differences between Session Beans and Message-Driven Beans, The Contents of an	<b>15L</b>



	Enterprise Bean, Naming Conventions for Enterprise Beans, The Life Cycles of Enterprise Beans, The Life Cycle of a Stateful Session Bean, The Life Cycle of a Stateless Session Bean, The Life Cycle of a Message-Driven Bean	
<b>Unit IV</b>	<b>Web Service:</b> Defining Client Access with Interfaces: Remote Access, Local Access, Local Interfaces and Container-Managed Relationships, Deciding on Remote or Local Access, Web Service Clients, Method Parameters and Access. Building Web Services with JAX-WS: Setting the Port, Creating a Simple Web Service and Client with JAX-WS.	<b>15L</b>
<b>Text book(s):</b>		
<ol style="list-style-type: none"> <li>1) Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD)</li> <li>2) Eric Jendrock, Jennifer Ball, D Carson and others, The Java EE 5 Tutorial, Pearson Education</li> <li>3) Bryan Basham, Kathy Sierra, Bert Bates, Head First Servlets and JSP, O'reilly (SPD)</li> </ol>		
<b>Additional Reference(s):</b>		
<ol style="list-style-type: none"> <li>1) Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR, 2001</li> <li>2) Ivan Bayross, Web Enabled Commercial Applications Development Using Java 2, BPB Publications</li> <li>3) The Java Tutorials of Sun Microsystems Inc.</li> </ol>		

<b>Course:</b> <b>USCS603</b>	<b>TOPICS (Credits : 2.5 Lectures/Week: 04)</b> <b>Software Engineering and Testing</b>	
<b>Unit I</b>	<b>Introduction to Software Engineering:</b> Introduction to Software, Types of Software , Classes of Software, Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Myths, Software Applications, Software-Engineering Processes, Evolution of Software,	<b>15L</b>

	<p>Comparison of Software Engineering and Related Fields, Some Terminologies, Programs Versus Software Products</p> <p><b>Software-Development Life-Cycle Models</b></p> <p>Software-Development Life-Cycle, Waterfall Model, Prototyping Model, Spiral Model, Evolutionary Development Model, Iterative-Enhancement Model, RAD Model, Comparison of Various Process, Models</p>	
<b>Unit II</b>	<p><b>Introduction to Software Requirements Specifications</b></p> <p>Requirement Engineering, Process of Requirements Engineering, Information Modeling, Data-Flow Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS Documents, SRS Validation, Components of SRS, Characteristics of SRS, Entity-Relationship Diagram</p> <p><b>Software Reliability and Quality Assurance</b></p> <p>Verification and Validation, Software Quality Assurance, Software Quality, (insert 6 sigma, Intro Agile Development) Capability Maturity Model (SEI-CMM), International Standard Organization (ISO), Comparison of ISO-9000 Certification and the SEI-CMM, Reliability Issues, Reliability Metrics, Reliability Growth Modeling, Reliability Assessment</p>	<b>15L</b>
<b>Unit III</b>	<p><b>System Design:</b> System/Software Design, Architectural Design, Low-Level Design</p> <p>Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design,</p> <p>Monitoring and Control for Design</p> <p><b>Software Measurement and Metrics:</b> Software Metrics, Halstead's Software Science, Function-Point Based Measures, Cyclomatic Complexity</p> <p><b>Software Testing :</b> Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box</p>	<b>15L</b>

	Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design	
<b>Unit IV</b>	<p><b>Software-Testing Strategies:</b> Static-Testing Strategies, Debugging, Error, Fault, and Failure</p> <p><b>Computer-Aided Software Engineering:</b> CASE and its Scope, Levels, Architecture of CASE Environment, Building Blocks, Support in Software Life-Cycle, Objectives, CASE Repository, Characteristics of CASE Tools, CASE Classification, Categories of CASE Tools, Advantages, Disadvantages of Case Tools, Reverse Software, Engineering, Software Re-Engineering</p> <p><b>Coding:</b> Information Hiding, Programming Style, Internal Documentation, Monitoring and Control for Coding, Structured Programming, Fourth-Generation Techniques</p>	<b>15L</b>
<p><b>Text book(s):</b></p> <ol style="list-style-type: none"> <li>1) Software Engineering, A Practitioner's Approach, Roger S, Pressman.</li> <li>2) Software Engineering, Ian Sommerville, Pearson Education</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1) Software Engineering Fundamentals, Behforooz, Hudson, Oxford University Press</li> <li>2) Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI</li> <li>3) Software Engineering-Principles and Practices, Jain, Oxford University Press</li> <li>4) Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley &amp; Sons</li> <li>5) Software Engineering Concepts, Richard Fairley, McGraw-Hill Companies</li> </ol>		

<b>Course:</b> <b>USCS604</b>	<b>TOPICS (Credits : 2.5 Lectures/Week: 04)</b> <b>Data Management using PL/SQL-II</b>	
<b>Unit I</b>	<b>Decomposition:</b> Functional dependency, Closure of a set of functional dependency, Lossless-Join decomposition, Multi valued dependency and fourth normal form, Join dependency, Fifth normal form.	<b>15L</b>

	<b>Concurrency Control:</b> Concept of a transaction, ACID properties, Serial and serializable schedules, Conflict and View serializabilty, Precedence graphs and test for conflict seralizability.	
<b>Unit II</b>	<b>Enforcing Serializability by locks:</b> Concept of locks, the locking scheduler, Two phase Locking, upgrading and down grading locks, Concept of deadlocks, Concurrency control by time stamps, The Thomos Write rule. <b>Crash Recovery:</b> ARIES algorithm. The log based recovery, recovery related structures like transaction and dirty page table, Write-ahead log protocol, check points, recovery from a system crash, Redo and Undo phases.	<b>15L</b>
<b>Unit III</b>	<b>Packages:</b> Advantages of Packages, Components of a Package, Develop a Package, Visibility of a Package’s components, Package Specification and Body, Package Constructs, PL/SQL Source Code Using the Data Dictionary <b>Dynamic SQL:</b> Execution Flow of SQL, Cursor Variables, Dynamically executing a PL/SQL Block, Configure Native Dynamic SQL to Compile PL/SQL Code, DBMS_SQL Package, Implement DBMS_SQL with a Parameterized DML Statement	<b>15L</b>
<b>Unit IV</b>	<b>Triggers:</b> Concepts of Triggers, Trigger Event Types and Body, Business Application Scenarios, Create Trigger, Insert Trigger and Delete Trigger Statement, Statement Level Triggers Versus Row Level Triggers, Create Instead of and Disabled Triggers, Managing Testing and Removing Triggers. <b>File Organization and Indexing:</b> Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes.	<b>15L</b>
<b>Text book(s):</b>		
<ol style="list-style-type: none"> <li>1) Ramakrishnam, Gehrke, “Database Management Systems”, McGraw- Hill.</li> <li>2) Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications</li> <li>3) Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner’s Guide, TataMcGraw-Hill.</li> </ol>		
<b>Additional Reference(s):</b>		
<ol style="list-style-type: none"> <li>1) Joel Murach, Murach’s MySQL, Mike Murach &amp; Associates</li> </ol>		

- 2) Elmasri and Navathe, “Fundamentals of Database Systems”, Pearson Education.
- 3) Peter Rob and Coronel, “Database Systems, Design, Implementation and Management”, Thomson Learning
- 4) ORACLE “The Complete Reference”, Tata McGraw Hill, New Delhi
- 5) C. J. Date, Longman, “Introduction to database Systems”, Pearson Education. George Koch and Kevin Loney

### **Suggested List of Practical – SEMESTER VI**

<b>Course:</b> <b>USCSP601</b>	<b>(Credits : 03 Practical/Week: 08)</b> <b>USCS602+USCS604</b>	
<b>Advanced JAVA Programming-II</b>		
<ol style="list-style-type: none"> <li>1. Simple Server-Side Programming using Servlets</li> <li>2. Advance Server-Side Programming using Servlets</li> <li>3. Simple Server-side programming using JSP</li> <li>4. Advance Server-side programming using JSP</li> <li>5. Developing Simple Enterprise Java Beans</li> <li>6. Developing Advance Enterprise Java Beans</li> <li>7. Developing Simple Web services in Java</li> <li>8. Developing Advance Web services in Java</li> </ol>		
<b>Data Management using PL/SQL-II</b>		
<ol style="list-style-type: none"> <li>1. Study of transactions and locks.</li> <li>2. Creating and Handling Deadlock situation.</li> <li>3. Packages 1: <ol style="list-style-type: none"> <li>a. Working with oracle supplied packages like DBMS_OUTPUT , etc</li> <li>b. Forward Declaration of packages</li> </ol> </li> <li>4. Packages 2: <ol style="list-style-type: none"> <li>a. Create and invoke a package that contains private and public constructs.</li> <li>b. Implement Package Functions in SQL</li> </ol> </li> <li>5. Data Dictionary: View PL/SQL Source Code Using the Data Dictionary.</li> </ol>		

- a. User Tables
  - b. All tables
  - c. DBA Tables
6. Dynamic SQL: Use of DBMS\_SQL package to write Dynamic SQL
- a. function and procedure of package (OPEN\_CURSOR, PARSE, BIND\_VARIABLE, EXECUTE, FETCH\_ROWS, CLOSE\_CURSOR)
  - b. Using the EXECUTE IMMEDIATE Statement
7. Dynamic SQL: Implementing DBMS\_SQL with a Parameterized DML Statement
8. Trigger: Creating and working with
- a. Insert/Update/Delete Trigger
  - b. Before/After Trigger
  - c. Working with statement Level Trigger and Row Level Trigger.
  - d. Remove Trigger
9. Indexes: Creating, dropping, and maintaining indexes on tables for the given column.

<b>USCSP602</b>	<b>(Credits: 03, Practical/Week: 08)</b>	
<b>USCS601+USCS603</b>		
<b>Project Documentation</b>		
<ol style="list-style-type: none"> <li><b>1. Acknowledgement</b></li> <li><b>2. Preliminary Investigation</b> - Organizational Overview, Description of System, Limitations of present system, Proposed system and its adv. [For web project, URL can be mentioned], Feasibility Study, Stakeholders, Technologies used, Gantt Chart</li> <li><b>3. System Analysis</b> - Fact Finding Techniques (Questionnaire, Sample Reports, Forms...), Prototypes(if any), Event Table, Use Case Diagram, Scenarios &amp; Use Case Description, ERD, Activity Diagram, Class diagram, Object Diagram, Sequence diagram/Collaboration Diagram, State diagram</li> <li><b>4. System Design</b> - Converting ERD to Tables, Design Class diagram[with UI classes, Persistent classes etc...], Component Diagram, Package Diagram, Deployment Diagram</li> <li><b>5. System Coding</b>- Menu Tree / Sitemap, List of tables with attributes and constraints, Design Patterns used (if any), Program Descr[ Programs /Classes and their responsibilities in brief ]</li> </ol>		

with Naming Conventions, Validations, Test Cases, Test Data and Test Results [Write test cases for all important programs], Screen Layouts & Report Layouts, Program Listing[for dummy project]

**6. System Implementation / Uploading**

**7. Future Enhancements**

**8. References and Bibliography**

**Note – Project documentation will carry 50 marks. They will be distributed as follows –**

1. Preliminary Investigation – 10 marks
2. System Analysis – 10 marks
3. System Design – 10 marks
4. System Coding & Implementation – 20 marks

**Project Development**

1. Faculties should arrange project demos for SY students at the end of the year or just at the beginning of TY. The demos can be of some good students of previous TY batches or it can be a project developed by faculties themselves.
2. SY students should be encouraged to start finding projects in the vacation. Faculties may take one or two introductory sessions for SY students before the vacation which will help students to work on preliminary investigation phase during vacation.
3. It can be Stand Alone, Multi-user or Web Based. Projects can be done in **any technology** and should have data stored in **DBMS**.
4. Each student shall do the project **individually**, though a project with the same topic name could be done by more than one student.
5. A project guide should be assigned to students. He/she will assign a schedule for each phase of the project and hand it over to students. The guides should oversee the project progress on a weekly/fortnightly basis. The guides should control iteration if any non-linear technique is used for project development.  
Sample phases can be as follows – Preliminary investigation, System Analysis, System Design, Coding, Implementation, Project Report Submission
6. College can arrange few sessions by experienced industry people on project management/best

practices/technologies etc.

7. After the completion of phase/projects, demos can be planned in front of faculties/clients/students.

**8. Projects should have at least following:**

- a. Good content management, presentation & meaningful images
  - b. Data Entry with Validations
  - c. Suitable navigation scheme(menus/toolbars/tabs/links etc)
  - d. Record Manipulation(add, update, delete, display, search ,sort)
  - e. Transactions / Sessions /Reports / Feedback/Registration whichever applicable
  - f. Login accounts(Admin & User) with separate functionalities for administrators and users
9. A certificate should be added in the project report which should contain the following information –
- a. The fact that the student has successfully completed the project as per the syllabus and that it forms a part of the requirements for completing the BSc degree in computer science of University of Mumbai.
  - b. The name of the student and the project guide
  - c. The academic year in which the project is done
  - d. Date of submission,
  - e. Signature of the project guide and the head of the department with date along with the department stamp,
  - f. Space for signature of the university examiner and date on which the project is evaluated.
10. Project should be evaluated by External Examiner as follows (Project Quality → 20 marks, Working of Project → 20 marks, Student's Presentation →10 marks)

**Note:**

- i. Evaluating “Project Quality”:** It involves overall modules included in the project, whether it was sufficiently large enough, whether validations were done for data entry, variety of reports etc.
- ii. Evaluating “Working of the Project”:** It involves error-free execution of the project.
- iii. Evaluating Student's Presentation:** Marks can be given based on the presentation skills of a student. A student can prepare a power point presentation for the project.





②

**UNIVERSITY OF MUMBAI**

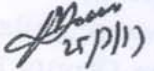
No. UG/109pf 2017-18

**CIRCULAR:-**

The Head of the University Departments of Computer Science, the Principals of the affiliated Colleges in Science and the Directors of recognized Science Institutions concerned are hereby informed that in continuation syllabi relating to Bachelor of Science degree Course passed by the Academic Council at its meeting held on 27/2/2013, vide item No. 4.51 and recommendations made by the Ad-hoc Board of Studies in Computer Science at its meeting held on 5/5/2017 has been accepted by the Academic Council at its meeting held on 11<sup>th</sup> May, 2017 vide item No. 4.212 and that in accordance therewith, the revised syllabus as per the (CBCS) for (Sem V & VI) of B.Sc programme in the Course of Computer Science, (Web Design and Technologies) which is available on the University's website ([www.mu.ac.in](http://www.mu.ac.in)) and that the same has been brought into force with effect from the academic year 2017-18.

MUMBAI - 400 032

27<sup>th</sup> July, 2017

  
REGISTRAR

To,

The Head of the University Departments of Computer Science, the Principals of the affiliated Colleges in Science and the Directors of Recognized Institutions concerned.

A.C/4.212/11.05.2017

No. UG/109-A of 2017

MUMBAI-400 032

27<sup>th</sup> July, 2017

Copy forwarded with compliments for information to :-

- 1) The Co-ordinator, Faculty of Science,
- 2) The Offg. Director, Board of Examinations and Evaluation,
- 3) The Director of Board of Studies Development,
- 4) The Professor-cum-Director, Institute of Distance and Open Learning.
- 5) The Co-Ordinator, University Computerization Centre.

  
REGISTRAR

....PTO

# **UNIVERSITY OF MUMBAI**



## **Syllabus for Semester V and VI**

**Program: B.Sc.**

**Course: Web Design and Technologies**

(Credit Based Semester and Grading System with  
effect from the academic year 2017–2018)

**T.Y.B.Sc. Computer Science Syllabus**  
**Credit Based Semester and Grading System**  
**To be implemented from the Academic year 2017-2018**

<b>SEMESTER V</b>			
<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>L / Week</b>
USACWD501	.Net Technologies	02	04
USACWD5P1	Practical of USACWD501	02	04

**THEORY**

<b>Course:</b>	<b>TOPICS (Credits : 02 Lectures/Week: 04)</b>	
<b>USACWD501</b>	<b>.Net Technologies</b>	
<b>Objectives:</b>		
To provide insight into .NET technologies for web programming and enable them design and develop interactive and responsive web applications. Explain learners the insights into the efficient usage of .NET technologies and their facilities.		
<b>Expected Learning Outcomes:</b>		
<ol style="list-style-type: none"> <li>1. To gain Knowledge of .NET technologies framework</li> <li>2. To understand and implement various controls for Creating a web Application</li> <li>3. Understand the security aspects of web Application.</li> </ol>		
<b>Unit I</b>	<b>Overview of .NET Framework</b> , Objectives, Main components of .NET Framework and their overview, Types of Applications .NET Framework Architecture– CLR(Goal of CLR, Services/Features, Benefits, Managed Execution Process, Automatic memory Management), CTS(CTS Overview, Type Definitions, Type members, Different types of data such as class, delegates, pointers, arrays, interfaces), Meta Data, Structure of Metadata & Self Describing Components, Cross Language Interoperability & CLS, Assemblies(Assembly overview, Benefits, Contents, Types)	<b>15L</b>

	<p><b>Creating Web Sites</b> :- Working with Web Forms, Working with CSS in Visual Web Developer, ASP.NET Server Controls, Standard Controls, HTML Controls, Understanding ASP.NET State Engine.</p>	
<b>Unit II</b>	<p><b>Introduction to Programming</b>:- Data Types and Variables, Statements, Methods: Functions and Subroutines.</p> <p>Consistent Page Layout with Master Pages, Using a Centralized Base Page.</p> <p><b>Structured Exception Handling</b> : try, catch, finally blocks, throwing exceptions, Err object, Using masked Textboxes</p> <p><b>Navigation Controls</b>- Architecture of the Navigation Controls, Menu Control, TreeView Control</p> <p><b>Validation Controls</b> – Validations &amp; Validator controls,</p>	<b>15L</b>
<b>Unit III</b>	<p><b>ADO.NET</b>: Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access</p> <p><b>Data Binding</b>: Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource, Other Data Controls, Working Together with Data Source and Data-bound Controls</p> <p><b>User Controls</b>-Creating User Controls, Adding User Controls to a Content Page or Master Page</p> <p><b>LINQ</b> -LINQ to Objects, LINQ to ADO.NET</p> <p>Introducing Query Syntax, Standard Query Operators, Shaping Data with Anonymous Types, Using Server Controls with LINQ Queries.</p>	<b>15L</b>
<b>Unit IV</b>	<p><b>Caching</b> : Introduction to Caching Data ,Different Ways to Cache Data in ASP.NET Web Applications</p> <p><b>Security</b>-Identity, Authentication, Authorization, Using Login Controls , Configuring the Web Application</p> <p><b>AJAX in ASP.NET</b> –Using ScriptManager, Partial refreshes, UpdatePanel, Triggers, Timed Refreshes</p> <p><b>Web Service</b> – What is web service, ASP.NET Web services, Creating a simple web service, Consuming Web service</p>	<b>15L</b>
<b>Textbook(s):</b>		

1. Beginning ASP.NET 4: in C# and VB, Imar Spaanjaars, WROX Publication
2. Murach's ASP.NET 4 Web Programming with C# 2010 by Anne Boehm , Joel Murach

**Additional Reference(s):**

1. The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill
2. Beginning ASP.NET 4 in C# and VB Imar Spanjaars, WROX

**Suggested List of Practical – SEMESTER V**

<b>Course:</b> <b>USACWD5P1</b>	<b>(Credits : 02 Practical/Week:04)</b> <b>Practical of USACWD501</b>	
<b>.Net Technologies</b>		
<ol style="list-style-type: none"> <li>1. Fundamentals of ASP.NET such as using HTML/Web Controls, Autopostback, Application events, Configuration files,</li> <li>2. ADO.NET Programming &amp; Data Binding.</li> <li>3. Creating Custom Controls.</li> <li>4. State management, Validation Controls</li> <li>5. LINQ- Using Server Controls with LINQ Queries</li> <li>6. Performance improvement using Caching</li> <li>7. Using AJAX in ASP.NET</li> <li>8. Creating &amp; Consuming web services</li> </ol>		

**T. Y. B. Sc.(Semester V and VI)**  
**Computer Science Syllabus**  
**Credit Based Semester and Grading System**  
**To be implemented from the Academic year 2017-2018**

<b>SEMESTER VI</b>			
<b>Course</b>	<b>TOPICS</b>	<b>Credits</b>	<b>L / Week</b>
USACWD602	Advanced Web Technology	02	04
USACWD6P2	Practical of USACWD602	02	04

**THEORY**

<b>Course:</b> USACWD602	<b>TOPICS (Credits : 02 Lectures/Week: 04)</b> <b>Advanced Web Technology</b>	
<b>Objectives:</b> To enable learners master the advanced topics in web programming and enable them design and develop interactive and responsive web applications. Explain learners the insights into the efficient usage of various web technology products and their facilities.		
<b>Learning Outcomes:</b> <ol style="list-style-type: none"> <li>1. Learn about using XML for managing Data</li> <li>2. Learn about using Ajax programming</li> <li>3. Learn about PHP and jQuery for developing web applications</li> </ol>		
<b>Unit I</b>	<b>XML-</b> Introduction to XML, Comparing XML with HTML Describing the Structure of XML - Declaration, Elements, Attributes, Comments, CDATA, XML Entity References, Parsers ,Describing Document Type Definitions, <b>Using XSLT with XML</b> :xsl:template Element , xsl:apply-templates Element,xsl:import , xsl:include Element , Element,xsl:element Element, xsl:attribute Element, xsl:value-of Element, using Conditional Statements, Sorting Elements, XSLT	<b>15L</b>

	functions, Creating Well-formed and Valid Documents.	
<b>Unit II</b>	<p><b>Introduction to Ajax –</b> AJAX Web Application Model, Working of AJAX</p> <p><b>Asynchronous Data Transfer with XMLHttpRequest</b>-Creating the XMLHttpRequest Object, XMLHttpRequest Properties, XMLHttpRequest Methods, Using the XMLHttpRequest Object in Different Browsers, Reading a File Synchronously, Reading a File Asynchronously, Performing Tasks Using the XMLHttpRequest Object,</p> <p><b>Integrating PHP and AJAX</b>-Sending Data from a Web Application to a Server, Validating a Field Using AJAX and PHP,</p>	<b>15L</b>
<b>Unit III</b>	<p><b>Handling XML Data using PHP and AJAX</b>-JavaScript properties for Extracting with nodeValue, Accessing XML Elements by Name, Accessing Attribute Values in XML Elements. Validating XML Documents in Ajax Applications</p> <p><b>Retrieving Data from a Database Using PHP and AJAX</b></p> <p><b>Consuming Web Services Using AJAX</b>-Exploring Web Service Protocols-SOAP, Web Service Description Language, UDDI, REST, Consuming Web Services Using AJAX</p>	<b>15L</b>
<b>Unit IV</b>	<p><b>jQuery-</b> JavaScript DOM objects their methods and properties-Window, History, Location Document, Form etc.</p> <p>Fundamentals of jQuery, Loading and using jQuery, using jQuery Library files, Callback functions, jQuery Selectors , jQuery Methods to Access HTML Attributes, jQuery Methods of traversing, jQuery Manipulators, jQuery Events, jQuery Effects, jQuery with AJAX</p>	<b>15L</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1) XML: A Beginner's Guide by Steven Holzner</li> <li>2) AJAX For Beginners , Ivan Bayross and Sharanam Shah, SPD</li> </ol>		



- 3) Web Development with jQuery (WROX) by Richard York
- 4) Learning PHP, MySQL & JavaScript with jQuery, CSS & HTML5 – by Robin Nixon, SPD

**Additional Reference(s):**

- 1) Ajax in Action Dave Crane, Eric Pascarello, Darren James
- 2) Ajax for Dummies Steve Holzner, PhD, Wiley Publishing Inc.

**Suggested List of Practical – SEMESTER VI**

<b>Course:</b> <b>USACWD6P2</b>	<b>(Credits : 02 Practical/Week:04)</b> <b>Practical of USACWD602</b>	
<b>Advanced Web Technology</b>		
<ol style="list-style-type: none"> <li>1. Representing Data using XML with XSL and Internal DTD</li> <li>2. Representing Data using XML with XSL and External DTD</li> <li>3. Retrieving data from server &amp; Sending data to server using AJAX</li> <li>4. Retrieving data from HTML form using PHP</li> <li>5. Retrieving Employee Details/ Registration Details from the database using Ajax and PHP</li> <li>6. Adding ,Modifying and Deleting data from Client side to into table in MySql Ajax and PHP</li> <li>7. Representing Data Using jQuery Selectors/ jQuery Methods to Access HTML Attributes</li> <li>8. Representing Data using jQuery Manipulators, jQuery Events, jQuery Effects</li> </ol>		

\*\*\*\*\*

# UNIVERSITY OF MUMBAI



**Syllabus for**  
**Program: Bachelor of Science**  
**Course: Computer Science**  
**Choice Based Credit System (CBCS)**  
**(Revised)**

*with effect from*  
**Academic Year 2021-2022**

## Preamble

The rise of Information and Communication Technology (ICT) has profoundly affected modern society. Increasing applications of computers in almost all areas of human endeavor has led to vibrant industries with concurrent rapid change in technology.

As the computing field advances at a rapid pace, the students must possess a solid foundation that allows and encourages them to maintain relevant skills as the field evolves. Specific languages and technology platforms change over time. Thus students must continue to learn and adapt their skills throughout their careers. To develop this ability, students will be exposed to multiple programming languages, tools, paradigms and technologies as well as the fundamental underlying principles throughout this programme.

The programme offers required courses such as programming languages, data structures, computer architecture and organization, algorithms, database systems, operating systems, and software engineering; as well as specialized courses in artificial intelligence, computer-based communication networks, distributed computing, information security, graphics, human-computer interaction, multimedia, scientific computing, web technology, and other current topics in computer science.

The core philosophy of this programme is to –

- Form strong foundations of Computer Science
- Nurture programming, analytical & design skills for the real world problems.
- Introduce emerging trends to the students in gradual way.
- Groom the students for the challenges of ICT industry

The students these days not only aspire for a career in the industry but also look for research opportunities. The main aim of this programme is to deliver a modern curriculum that will equip graduates with strong theoretical and practical backgrounds to enable them to excel in the workplace and to be lifelong learners. Not only does it prepares the students for a career in Software industry, it also motivates them towards further studies and research opportunities. Graduating students, can thus take up postgraduate programmes in CS leading to research as well as R&D, can be employable at IT industries, or can adopt a business management career.

In the first year i.e. for semester I & II, basic foundation of important skills required for software development is laid. The syllabus proposes to have four core subjects of Computer science and two core courses of Mathematics-Statistics. All core subjects are proposed to have theory as well as practical tracks. While the Computer Science courses will form fundamental skills for solving computational problems, the Mathematics & Statistics course will inculcate research-oriented acumen. Ability Enhancement Courses on Soft Skill Development will ensure an overall and holistic development of the students. The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science.

We sincerely believe that any student taking this programme will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students' community and teachers' fraternity will appreciate the treatment given to the courses in the syllabus.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents, we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly helped designing certain specialized courses and the syllabus as a whole.

## Programme Structure for B.Sc. Computer Science

Programme Duration	06 Semesters <i>spread across 3 years</i>
Total Credits required for successful completion of the Course	120
Credits required from the Core Courses	76
Credits required for the Ability Enhancement Courses	04
Credits required for Skills Enhancement Courses	32
Credits for General Elective Courses	08
Minimum Attendance per Semester	75%

### Programme Objectives

**The objectives of the 3 year B.Sc. Computer Science programme are as follows:**

- To develop an understanding and knowledge of the basic theory of Computer Science with good foundation on theory, systems and applications.
- To foster necessary skills and analytical abilities for developing computer based solutions of real-life problems.
- To provide training in emergent computing technologies which lead to innovative solutions for industry and academia.
- To develop the necessary study skills and knowledge to pursue further post-graduate study in computer science or other related fields.
- To develop the professional skillset required for a career in an information technology oriented business or industry.
- To enable students to work independently and collaboratively, communicate effectively, and become responsible, competent, confident, insightful, and creative users of computing technology

## **Programme Learning Outcomes**

**At the end of three year Bachelor of Computer Science the students will be able:**

- To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems.
- To design and develop computer programs/computer -based systems in the areas such as networking, web design, security, cloud computing, IoT, data science and other emerging technologies.
- To familiarize with the modern-day trends in industry and research based settings and thereby innovate novel solutions to existing problems.
- To apply concepts, principles, and theories relating to computer science to new situations.
- To use current techniques, skills, and tools necessary for computing practice
- To apply standard Software Engineering practices and strategies in real-time software project development
- To pursue higher studies of specialization and to take up technical employment.
- To work independently or collaboratively as an effective team member on a substantial software project.
- To communicate and present their work effectively and coherently.
- To display ethical code of conduct in usage of Internet and Cyber systems.
- To engage in independent and life-long learning in the background of rapid changing IT industry.

# F.Y.B.Sc. Computer Science Syllabus

## Choice Based Credit System (CBCS)

*with effect from*

**Academic year 2021-2022**

<b>Semester – I</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
USCS101	Core Subject	Digital Systems & Architecture	2	3
USCSP101	Core Subject Practical	Digital Systems & Architecture – Practical	1	3
USCS102	Core Subject	Introduction to Programming with Python	2	3
USCSP102	Core Subject Practical	Introduction to Programming with Python – Practical	1	3
USCS103	Core Subject	LINUX Operating System	2	3
USCSP103	Core Subject Practical	LINUX Operating System – Practical	1	3
USCS104	Core Subject	Open Source Technologies	2	3
USCSP104	Core Subject Practical	Open Source Technologies – Practical	1	3
USCS105	Core Subject	Discrete Mathematics	2	3
USCSP105	Core Subject Practical	Discrete Mathematics – Practical	1	3
USCS106	Core Subject	Descriptive Statistics	2	3
USCSP106	Core Subject Practical	Descriptive Statistics – Practical	1	3
USCS107	Ability Enhancement Course	Soft Skills	2	3

# F.Y.B.Sc. Computer Science Syllabus

## Choice Based Credit System (CBCS)

*with effect from*

**Academic year 2021-2022**

<b>Semester – II</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
USCS201	Core Subject	Design & Analysis of Algorithms	2	3
USCSP201	Core Subject Practical	Design & Analysis of Algorithms – Practical	1	3
USCS202	Core Subject	Advanced Python Programming	2	3
USCSP202	Core Subject Practical	Advanced Python Programming – Practical	1	3
USCS203	Core Subject	Introduction to OOPs using C++	2	3
USCSP203	Core Subject Practical	Introduction to OOPs using C++ – Practical	1	3
USCS204	Core Subject	Database Systems	2	3
USCSP204	Core Subject Practical	Database Systems – Practical	1	3
USCS205	Core Subject	Calculus	2	3
USCSP205	Core Subject Practical	Calculus – Practical	1	3
USCS206	Core Subject	Statistical Methods	2	3
USCSP206	Core Subject Practical	Statistical Methods – Practical	1	3
USCS207	Ability Enhancement Course	E-Commerce & Digital Marketing	2	3

## Semester I

Course Code	Course Title	Credits	Lectures /Week
USCS101	Digital Systems & Architecture	2	3
<p><b>About the Course:</b>            This course introduces the principles of computer organization and the basic architecture concepts. The course emphasizes performance and cost analysis, instruction set design, pipelining, memory technology, memory hierarchy, virtual memory management, and I/O systems.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To have an understanding of Digital systems and operation of a digital computer.</li> <li>To learn different architectures &amp; organizations of memory systems, processor organization and control unit.</li> <li>To understand the working principles of multiprocessor and parallel organization's as advanced computer architectures</li> </ul>			
<p><b>Learning Outcomes:</b>            After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>To learn about how computer systems work and underlying principles</li> <li>To understand the basics of digital electronics needed for computers</li> <li>To understand the basics of instruction set architecture for reduced and complex instruction sets</li> <li>To understand the basics of processor structure and operation</li> <li>To understand how data is transferred between the processor and I/O devices</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Fundamentals of Digital Logic:</b> Boolean algebra, Logic Gates, Simplification of Logic Circuits: Algebraic Simplification, Karnaugh Maps. Combinational Circuits: Adders, Mux, De-Mux, Sequential Circuits: Flip-Flops (SR, JK &amp; D), Counters: synchronous and asynchronous Counter</p> <p><b>Computer System:</b> Comparison of Computer Organization &amp; Architecture, Computer Components and Functions, Interconnection Structures. Bus Interconnections, Input / Output: I/O Module, Programmed I/O, Interrupt Driven I/O, Direct Memory Access</p>	15	
II	<p><b>Memory System Organization:</b> Classification and design parameters, Memory Hierarchy, Internal Memory: RAM, SRAM and DRAM, Interleaved and Associative Memory. Cache Memory: Design Principles, Memory mappings, Replacement Algorithms, Cache performance, Cache Coherence. Virtual Memory, External Memory: Magnetic Discs, Optical Memory, Flash Memories, RAID Levels</p> <p><b>Processor Organization:</b> Instruction Formats, Instruction Sets, Addressing Modes, Addressing Modes Examples with Assembly Language [8085/8086 CPU], Processor Organization, Structure and Function. Register</p>	15	



	Organization, Basic Microprocessor operations: Data Transfer (Register / Memory) Operations, Arithmetic & Logical Operations, Instruction Cycle, Instruction Pipelining. Introduction to RISC and CISC Architecture, Instruction Level Parallelism and Superscalar Processors: Design Issues	
<b>III</b>	<p><b>Control Unit:</b> Micro-Operations, Functional Requirements, Processor Control, Hardwired Implementation, Micro-programmed Control.</p> <p><b>Fundamentals of Advanced Computer Architecture:</b> Parallel Architecture: Classification of Parallel Systems, Flynn's Taxonomy, Array Processors, Clusters, and NUMA Computers. Multiprocessor Systems: Structure &amp; Interconnection Networks, Multi-Core Computers: Introduction, Organization and Performance.</p>	<b>15</b>

**Textbooks:**

1. M. Mano, Computer System Architecture 3rd edition, Pearson
2. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012
3. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010

**Additional References:**

1. William Stallings (2010), Computer Organization and Architecture- designing for performance, 8th edition, Prentice Hall, New Jersey.
2. Anrew S. Tanenbaum (2006), Structured Computer Organization, 5th edition, Pearson Education Inc,
3. John P. Hayes (1998), Computer Architecture and Organization, 3rd edition, Tata McGrawHill

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP101</b>	<b>Digital Systems &amp; Architecture – Practical</b>	<b>1</b>	<b>3</b>
1	Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).		
2	Simplify given Boolean expression and realize it.		
3	Design and verify a half/full adder		
4	Design and verify half/full subtractor		
5	Design a 4 bit magnitude comparator using combinational circuits.		
6	Design and verify the operation of flip-flops using logic gates.		
7	Verify the operation of a counter.		
8	Verify the operation of a 4 bit shift register		
9	Design and implement expression using multiplexers / demultiplexers.		
10	Design and implement 3-bit binary ripple counter using JK flip flops.		
11	Simple microprocessor programs for data transfer operations		
12	Simple microprocessor programs for arithmetic & logical transfer operations		
Note	Practical 1 – 10 can be performed using any open source simulator (like Logisim) (Download it from <a href="https://sourceforge.net/projects/circuit/">https://sourceforge.net/projects/circuit/</a> )		

	Practical 11 – 12 can be performed on any simulation software like Jubin’s 8085 simulator		
<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCS102</b>	<b>Introduction to Programming with Python</b>	<b>2</b>	<b>3</b>
<b>About the Course:</b>			
This course is aims at introducing one of the fastest growing programming language of current time and enables learners to understand the fundamentals of programming with Python. Learners will be able to write programs to solve real-world problems, and produce quality code. It will help to develop strong skills of programming for implementing applications for emerging fields including data science and machine learning.			
<b>Course Objectives:</b>			
<ul style="list-style-type: none"> <li>• To learn how to design and program Python applications.</li> <li>• To explore the innards of Python Programming and understand components of Python Program</li> <li>• To define the structure and components of a Python program.</li> <li>• To learn how to write loops and decision statements in Python</li> <li>• To learn about inbuilt input/output operations and compound data types in Python</li> </ul>			
<b>Learning Outcomes:</b>			
After successful completion of this course, students would be able to:			
<ul style="list-style-type: none"> <li>• Ability to store, manipulate and access data in Python</li> <li>• Ability to implement basic Input / Output operations in Python</li> <li>• Ability to define the structure and components of a Python program.</li> <li>• Ability to learn how to write loops and decision statements in Python.</li> <li>• Ability to learn how to write functions and pass arguments in Python.</li> <li>• Ability to create and use Compound data types in Python</li> </ul>			
<b>Unit</b>	<b>Topics</b>	<b>No of Lectures</b>	
<b>I</b>	<p><b>Overview of Python:</b> History &amp; Versions, Features of Python, Execution of a Python Program, Flavours of Python, Innards of Python, Python Interpreter, Memory Management in Python, Garbage Collection in Python, Comparison of Python with C and Java, Installing Python, Writing and Executing First Python Program, Getting Help, IDLE</p> <p><b>Data Types, Variables and Other Basic Elements:</b> Comments, Docstrings, Data types- Numeric Data type, Compound Data Type, Boolean Data type, Dictionary, Sets, Mapping, Basic Elements of Python, Variables</p> <p><b>Input and Output Operations:</b> Input Function, Output Statements, The print() function, The print(“string”) function, The print(variables list) function, , The print(object) function, The print(formatted string) function, Command Line Arguments</p>	<b>15</b>	

	<p><b>Control Statements:</b> The if statement, The if ... else Statement, The ‘if ... elif ... else’ Statement, Loop Statement- while loop, for loop, Infinite loop, Nested loop, The else suite, break statement, continue statement, pass statement, assert statement, return statement</p>	
<b>II</b>	<p><b>Operators:</b> Arithmetic operators, Assignment operators, Unary minus operator, Relational operators, Logical operators, Bitwise operators, Membership operators, Identity operators, Precedence of Operators, Associativity of Operators</p> <p><b>Arrays:</b> Creating Arrays, Indexing and Slicing of Arrays, Basic Array Operations, Arrays Processing, Mathematical Operations on Array, Aliasing Arrays, Slicing and Indexing in NumPy Arrays, Basic slicing, Advanced Indexing, Dimensions of Arrays, Attributes of an Array, The ndim Attribute, The shape Attribute, The size Attribute, The itemsize Attribute</p> <p><b>Functions:</b> Function definition and call, Returning Results, Returning Multiple Values from a Function, Built-in Functions, Difference between a Function and a Method, Pass Value by Object Reference, Parameters and Arguments, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Arbitrary Arguments, Recursive Functions, Anonymous or Lambda Functions, Using Lambda with the filter() Function, Using Lambda with the map() Function, Using Lambda with the reduce() Function</p> <p><b>Modules:</b> Introduction to Modules in Python</p>	<b>15</b>
<b>III</b>	<p><b>Strings:</b> Creating Strings, Functions of Strings, Working with Strings, Length of a String, Indexing and Slicing, Repeating and Concatenating Strings, Checking Membership, Comparing Strings, Removing Spaces, Finding Substrings, Counting Substrings, Immutability, Splitting and Joining Strings, Changing Case, Checking Starting and Ending of a String, Sorting Strings, Searching in the Strings, Testing Methods, Formatting Strings, Finding the Number of Characters and Words, Inserting Substrings into a String</p> <p><b>List and Tuples:</b> Lists, List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples, Functions in Tuple</p> <p><b>Dictionaries:</b> Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries, Converting Lists into Dictionary, Converting Strings into Dictionary, Passing Dictionaries to Functions, Sorting the Elements of a Dictionary using Lambda, Ordered Dictionaries</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition, 2014</li> </ol>		

2. Programming through Python, M. T Savaliya, R. K. Maurya & G M Magar, Sybgen Learning India, 2020

**Additional References:**

1. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018
2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017
3. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018
4. Python Programming: Using Problem Solving Approach, Reema Thareja, Oxford University Press, 2017
5. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP102</b>	<b>Introduction to Programming with Python – Practical</b>	<b>1</b>	<b>3</b>
1	Write a program to design and develop python program to implement various control statement using suitable examples		
2	Write program in Python to define and call functions for suitable problem.		
3	Write Python program to demonstrate different types of function arguments.		
4	Write a Python program to demonstrate the precedence and associativity of operators.		
5	Write suitable Python program to implement recursion for problems such as Fibonacci series, Factorial, Tower of Hanoi etc.		
6	Write Python program to implement and use lambda function in python		
7	Write a python program to create and manipulate arrays in Python. Also demonstrate use of slicing and indexing for accessing elements from the array.		
8	Write a program to implement list in Python for suitable problem. Demonstrate various operations on it.		
9	Write a program to implement tuple in Python for suitable problem. Demonstrate various operations on it.		
10	Write a program to implement dictionary in Python for suitable problem. Demonstrate various operations on it.		

Course Code	Course Title	Credits	Lectures /Week
USCS103	LINUX Operating System	2	3
<p><b>About the Course:</b>  This syllabus will help to train students in fundamental skills and build-up sustainable interest in Linux Operating System. It will improve necessary knowledge base to understand Linux Operating System and its practical implementation, it will also help to develop Linux based solutions for real life problems.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To learn basic concepts of Linux in terms of operating system</li> <li>To learn use of various shell commands with regular expressions</li> <li>To set Linux Environment variables and learn setting file permissions to maintain Linux security implementation</li> <li>To learn various editors available in Linux OS</li> <li>To learn shell scripting.</li> <li>To learn installation of compilers and programming using C and Python languages on Linux platform</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Work with Linux file system structure, Linux Environment</li> <li>Handle shell commands for scripting, with features of regular expressions, redirections</li> <li>Implement file security permissions</li> <li>Work with vi, sed and awk editors for shell scripting using various control structures</li> <li>Install softwares like compilers and develop programs in C and Python programming languages on Linux Platform</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Linux operating system and Basics :</b> History, GNU Info and Utilities, Various Linux Distributions, The Unix/Linux architecture, Features of Unix/Linux, Starting the shell, Shell prompt, Command structure, File Systems and Directory Structure, man pages, more documentation pages</p> <p><b>Basic Bash shell commands:</b> General purpose utility Commands, basic commands, Various file types, attributes and File handling Commands, Handling Ordinary Files. More file attributes</p> <p><b>Advanced Bash shell commands:</b>Simple Filters, Filters using regular expressions.</p> <p><b>The Linux environment variable:</b> Setting, Locating and removing environment variables like PATH etc, Default shell environment variables, Using command aliases.</p>	15	

<b>II</b>	<p><b>Understanding Linux file permission:</b> Linux security, Using Linux groups, Decoding file permissions, Changing security setting, Sharing files.</p> <p><b>Linux Security:</b> Understanding Linux Security, uses of root, sudo command, working with passwords, Understanding ssh.</p> <p><b>Networking:</b> TCP/IP Basics, TCP/IP Model, Resolving IP addresses, Applications, ping, telnet, ftp, DNS</p> <p><b>Working with Editors:</b> awk, sed and Introduction to vi</p>	<b>15</b>
<b>III</b>	<p><b>Basic script building:</b> Using multiple commands, Creating script files, Displaying messages, Using variables, Redirecting Input and Output, Pipes performing math, Exiting the script.</p> <p><b>Using structured commands:</b> Working with if-then, if-then-else and nested if statements, test command, Compound condition testing, while command, until command, case command.</p> <p><b>Script and Process control :</b> Handling signals, Running scripts in background mode, Running scripts without a console, Job control, Job scheduling commands: ps, nice, renice, at, batch, cron table, Running the script at boot</p>	<b>15</b>

**Textbooks:**

1. “Linux Command line and Shell Scripting Bible”, Richard Blum, Wiley India.
2. “Unix: Concepts and Applications”, Sumitabha Das, 4th Edition, McGraw Hill.
3. “Official Ubuntu Book”, Matthew Helmke& Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, 8th Ed.

**Additional References:**

1. “Linux Administration: A Beginner's Guide”, Fifth Edition, Wale Soyinka, Tata McGraw-Hill, 2008.
2. “Linux: Complete Reference”, Richard Petersen, 6th Edition, Tata McGraw-Hill
3. “Beginning Linux Programming”, Neil Mathew, 4th Edition, Wiley Publishing, 2008.

Course Code	Course Title	Credits	Lectures /Week
USCSP103	<b>LINUX Operating System – Practical</b>	<b>1</b>	<b>3</b>
1	<p><b>Installation of Ubuntu Linux operating system.</b></p> <p>a) Booting and Installing from ( USB/DVD)</p> <p>b) Using Ubuntu Software center / Using Synaptic</p> <p>c) Explore useful software packages.</p>		
2	<p><b>Becoming an Ubuntu power user</b></p> <p>a) Administering system and User setting</p> <p>b) Learning Unity keyboard</p> <p>c) Using the Terminal</p> <p>d) Working with windows programs</p>		

3	<p><b>File System Commands:</b> touch, help, man, more, less, pwd, cd, mkdir, rmdir, ls, find, ls, etc</p> <p><b>File handling Commands:</b> cat, cp, rm, mv, more, file, wc, od, cmp, diff, comm, chmod, chown, chgrp, gzip and gunzip, zip and unzip, tar, ln, umask,, chmod, chgrp, chown, etc</p>
4	<p><b>General purpose utility Commands:</b>cal, date, echo, man, printf, passwd, script, who, uname, tty, stty, etc</p> <p><b>Simple Filters and I/O redirection:</b> head, tail, cut paste, sort, grep family, tee, uniq, tr, etc.</p> <p><b>Networking Commands:</b> who, whoami, ping, telnet, ftp, ssh, etc</p>
5	<p><b>Editors:</b> vi, sed, awk</p>
6	<p><b>Working and Managing with processes-</b> sh, ps, kill, nice, at and batch etc.</p>
7	<p><b>Shell scripting I:</b> Defining variables, reading user input, exit and exit status commands, , expr, test, [], if conditional, logical operators</p>
8	<p><b>Shell scripting II:</b> Conditions (for loop, until loop and while loop) arithmetic operations, examples</p>
9	<p><b>Shell scripting III:</b> Redirecting Input / Output in scripts, creating your own Redirection</p>
10	<p>Installation of C/C++/Java/Python Compiler and creating an environment for app development. Basic programming using C and Python Languages.</p>

Course Code	Course Title	Credits	Lectures /Week
USCS104	Open Source Technologies	2	3
<p><b>About the Course:</b>  Open Source Software is becoming an important resource for development, especially in developing countries. A working understanding of the economic and technical background of the Free / Open Source Software movement (FOSS) is essential for its effective use. The course takes students through the history and current status of the FOSS world, and starts them exploring it, by connecting their personal experiences with corresponding FOSS projects. Students will experience finding and using Open Source Software projects.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the difference between open-source software and commercial software.</li> <li>• Understand the policies, licensing procedures and ethics of FOSS.</li> <li>• Understand open-source philosophy, methodology and ecosystem.</li> <li>• Awareness with Open-Source Technologies.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <ul style="list-style-type: none"> <li>• Differentiate between Open Source and Proprietary software and Licensing.</li> <li>• Recognize the applications, benefits and features of Open-Source Technologies</li> <li>• Gain knowledge to start, manage open-source projects.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Open-Source:</b> Open Source, Need and Principles of OSS, Open-Source Standards, Requirements for Software, OSS success, Free Software, Examples, Licensing, Free Vs. Proprietary Software, Free Software Vs. Open-Source Software, Public Domain. History of free software, Proprietary Vs Open-Source Licensing Model, use of Open-Source Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project.</p> <p><b>Open-Source Principles and Methodology:</b> Open-Source History, Open-Source Initiatives, Open Standards Principles, Methodologies, Philosophy, Software freedom, Open-Source Software Development, Licenses, Copyright vs. Copy left, Patents, Zero marginal cost, Income-generation Opportunities, Internationalization.</p> <p><b>Licensing:</b> What Is A License, How to create your own Licenses, Important FOSS Licenses (Apache, BSD, PL, LGPL), copyrights and copy lefts, Patent.</p>	15	
II	<p><b>Open-Source projects:</b> Starting and maintaining own Open-Source Project, Open-Source Hardware, Open-Source Design, Open-source Teaching, Open-source media.</p> <p><b>Collaboration:</b> Community and Communication, Contributing to Open-Source Projects Introduction to GitHub, interacting with the community on GitHub, Communication and etiquette, testing open-source code, reporting</p>	15	



	<p>issues, contributing code. Introduction to Wikipedia, contributing to Wikipedia or contributing to any prominent open-source project of student's choice.</p> <p><b>Open-Source Ethics and Social Impact:</b> Open source vs. closed source, Open-source Government, Ethics of Open-source, Social and Financial impacts of open-source technology, Shared software, Shared source, Open Source as a Business Strategy</p>	
<b>III</b>	<p><b>Understanding Open-Source Ecosystem:</b> Open-Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open-Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, Debuggers, Programming languages, LAMP, Open-Source Database technologies</p> <p><b>Case Studies:</b> Example Projects: Apache Web server, BSD, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, WordPress, Git, GCC, GDB, GitHub, Open Office, LibreOffice</p> <p><b>Study:</b> Understanding the developmental models, licensing, mode of funding, commercial/non-commercial use.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. "Open-Source Technology", Kailash Vadera&amp;Bhavyesh Gandhi, University Science Press, Laxmi Publications, 2009</li> <li>2. "Open-Source Technology and Policy", Fadi P. Deek and James A. M. McHugh, Cambridge University Press, 2008.</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. "Perspectives on Free and Open-Source Software", Clay Shirky and Michael Cusumano, MIT press.</li> <li>2. "Understanding Open Source and Free Software Licensing", Andrew M. St. Laurent, O'Reilly Media.</li> <li>3. "Open Source for the Enterprise", Dan Woods, GautamGuliani, O'Reilly Media</li> <li>4. Linux kernel Home: <a href="http://kernel.org">http://kernel.org</a></li> <li>5. Open-Source Initiative: <a href="https://opensource.org/5">https://opensource.org/5</a></li> <li>6. The Linux Foundation: <a href="http://www.linuxfoundation.org/">http://www.linuxfoundation.org/</a></li> <li>7. The Linux Documentation Project: <a href="http://www.tldp.org/2">http://www.tldp.org/2</a></li> <li>8. Docker Project Home: <a href="http://www.docker.com3">http://www.docker.com3</a>.</li> <li>9. Linux Documentation Project: <a href="http://www.tldp.org/6">http://www.tldp.org/6</a></li> <li>10. Wikipedia: <a href="https://en.wikipedia.org/7">https://en.wikipedia.org/7</a>.<a href="https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia8">https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia8</a></li> <li>11. GitHub: <a href="https://help.github.com/9">https://help.github.com/9</a>.</li> <li>12. The Linux Foundation: <a href="http://www.linuxfoundation.org/">http://www.linuxfoundation.org/</a></li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP104</b>	<b>Open Source Technologies– Practical</b>	<b>1</b>	<b>3</b>
1	<b>Open Source Operating Systems</b> <ul style="list-style-type: none"> <li>Learn the following open source operating system of your choice: Linux, Android, FreeBSD, Open Solaris etc.</li> <li>Learn the installation.</li> <li>Identify the unique features of these OS.</li> </ul>		
2	Hands on with <b>LibreOffice</b> <ul style="list-style-type: none"> <li>Learn it from practical view-point</li> <li>Give a brief presentation about it to the class</li> </ul>		
3	Hands on with <b>GIMP</b> Photo Editing Tool <ul style="list-style-type: none"> <li>Learn it from practical view-point</li> <li>Give a brief presentation about it to the class</li> </ul>		
4	Hands on with <b>Shotcut</b> Video Editing Tool <ul style="list-style-type: none"> <li>Learn it from practical view-point</li> <li>Give a brief presentation about it to the class</li> </ul>		
5	Hands on with <b>Blender</b> Graphics and Animation Tool <ul style="list-style-type: none"> <li>Learn it from practical view-point</li> <li>Give a brief presentation about it to the class</li> </ul>		
6	Hands on with <b>Apache</b> Web Server <ul style="list-style-type: none"> <li>Learn it from practical view-point</li> <li>Give a brief presentation about it to the class</li> </ul>		
7	Hands on with <b>WordPress</b> CMS <ul style="list-style-type: none"> <li>Learn it from practical view-point</li> <li>Give a brief presentation about it to the class</li> </ul>		
8	Contributing to <b>Wikipedia</b> : <ul style="list-style-type: none"> <li>Introduction to wikipedia: operating model, license, how to contribute?</li> <li>Create your user account on wikipedia</li> <li>c. Identify any topic of your choice and contribute the missing information</li> </ul>		
9	<b>Github</b> <ul style="list-style-type: none"> <li>Create and publish your own open source project: Write any simple program using your choice of programming language.</li> <li>Create a repository on github and save versions of your project. You'll learn about the staging area, committing your code, branching, and merging,</li> <li>Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi-developer projects. You'll learn how to make and review a pull request on GitHub.</li> <li>d. Contribute to a Live Project: Students will publish a repository containing</li> </ul>		

	their reflections from the course and submit a pull request.
10	<b>Virtualization:</b> Open Source virtualization technologies: <ul style="list-style-type: none"><li>• Install and configure the following: VirtualBox, Zen, KVM</li><li>• Create and use virtual machines</li></ul>
11	<b>Containerization:</b> <ul style="list-style-type: none"><li>• Install and configure the following containerization technologies: docker, rocket, LXD</li><li>• Create and use containers using it</li></ul>

Course Code	Course Title	Credits	Lectures /Week
USCS105	Discrete Mathematics	2	3
<p><b>About the Course:</b>  Discrete Mathematics provides an essential foundation for virtually every area of Computer Science. The problem-solving techniques honed in Discrete Mathematics are necessary for writing complicated software. Discrete mathematics also builds the gateway to advanced courses in Mathematical Sciences, Data Science, Machine Learning, Software Engineering, etc.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• The purpose of the course is to familiarize the prospective learners with mathematical structures that are fundamentally discrete.</li> <li>• This course will enhance prospective learners to reason and ability to articulate mathematical problems.</li> <li>• This course will introduce functions, forming and solving recurrence relations and different counting principles. These concepts will be useful to study or describe objects or problems in computer algorithms and programming languages and these concepts can be used effectively in other courses.</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, learners would be able to:</p> <ul style="list-style-type: none"> <li>• Define mathematical structures (relations, functions, graphs) and use them to model real life situations.</li> <li>• Understand, construct and solve simple mathematical problems.</li> <li>• Solve puzzles based on counting principles.</li> <li>• Provide basic knowledge about models of automata theory and the corresponding formal languages.</li> <li>• Develop an attitude to solve problems based on graphs and trees, which are widely used in software.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Functions:</b>  Definition of function; Domain, co-domain, range of a function; Examples of standard functions such as identity and constant functions, absolute value function, logarithmic and exponential functions, flooring and ceiling functions; Injective, surjective and bijective functions; Composite and inverse functions.</p> <p><b>Relations:</b>  Definition and examples of relation; Properties of relations, Representation of relations using diagraphs and matrices; Equivalence relation; Partial Order relation, Hasse Diagrams, maximal, minimal, greatest, least element, Lattices.</p>	15	

	<p><b>Recurrence Relations:</b>  Definition and Formulation of recurrence relations; Solution of a recurrence relation; Solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients; Homogeneous solution of linear homogeneous recurrence relation with constant coefficients; Particular solution of non-linear homogeneous recurrence relation with constant coefficients; General solution of non-linear homogeneous recurrence relation with constant coefficients; Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi, Intersection of lines in a plane, Sorting Algorithms.</p>	
II	<p><b>Counting Principles:</b>  Basic Counting Principles (Sum and Product Rule); Pigeonhole Principle (without proof) - Simple examples; Inclusion Exclusion Principle (Sieve formula) (without proof); Counting using Tree diagrams.</p> <p><b>Permutations and Combinations:</b>  Permutation without and with repetition; Combination without and with repetition; Binomial numbers and identities: Pascal Identity, Vandermonde’s Identity, Pascal triangle, Binomial theorem (without proof) and applications; Multinomial numbers, Multinomial theorem (without proof) and applications.</p> <p><b>Languages, Grammars and Machines:</b>  Languages and Grammars – Introduction, Phase structure grammar, Types of grammar, derivation trees; Finite-State Machines with Output; Finite-State Machines with No Output; Regular Expression and Regular Language.</p>	15
III	<p><b>Graphs:</b>  Graphs and Graph Models; Graph terminologies and Special types of graphs; Definition and elementary results; Representing graphs, Linked representation of a graph; Graph Isomorphism; Connectivity in graphs – path, trail, walk; Euler and Hamilton paths; Planar graphs, Graph coloring and chromatic number.</p> <p><b>Trees:</b>  Definition, Tree terminologies and elementary results; Linked representation of binary trees; Ordered rooted tree, Binary trees, Complete and extended binary trees, Expression trees, Binary Search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree; Traversing binary trees</p>	15
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)</li> <li>2. Discrete Mathematics: SemyourLipschutz, Marc Lipson, Schaum’s out lines, McGraw- Hill Inc.</li> </ol>		

3<sup>rd</sup> Edition

3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc. 2017
4. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.

**Additional References:**

1. Elements of Discrete Mathematics: C.L. Liu, Tata McGraw- Hill Edition.
2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
3. Discrete Mathematics: SemyourLipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc.
4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi.

Course Code	Course Title	Credits	Lectures /Week
USCSP105	Discrete Mathematics – Practical	1	3
1	<b>Functions –</b> <ol style="list-style-type: none"> <li>a. Identify if the given mapping is a function</li> <li>b. Finding domain and range of a given function</li> <li>c. Check if the given function is injective/surjective/bijective</li> <li>d. Find the inverse of a given function</li> <li>e. Operations on functions</li> <li>f. Graphs of functions using any online tool</li> </ol>		
2	<b>Relations –</b> <ol style="list-style-type: none"> <li>a. Representation of relations</li> <li>b. Determine if the given relation satisfies equivalence relation/partial order relation</li> <li>c. Draw Hasse diagrams</li> <li>d. Find maximal, minimal, greatest, least element in a poset</li> <li>e. Determine if a given poset is a lattice</li> </ol>		
3	<b>Recurrence Relation –</b> <ol style="list-style-type: none"> <li>a. Solve recurrence relation using backtracking method</li> <li>b. Solve linear homogeneous recurrence relations with constant coefficients</li> <li>c. Find homogeneous, particular, general solution of a recurrence relation</li> <li>d. Formulate and solving recurrence relation</li> </ol>		
4	<b>Counting Principles –</b> <ol style="list-style-type: none"> <li>a. Sum and product rule</li> <li>b. Pigeonhole Principle</li> <li>c. Inclusion Exclusion Principle</li> <li>d. Counting using Tree diagrams</li> </ol>		
5	<b>Permutations and Combinations –</b> <ol style="list-style-type: none"> <li>a. Permutations</li> <li>b. Permutations with repetitions</li> <li>c. Combinations</li> <li>d. Combinations with repetitions</li> <li>e. Binomial numbers and Identities</li> </ol>		

	<ul style="list-style-type: none"> <li>f. Applications on Binomial theorem</li> <li>g. Applications on Multinomial theorem</li> </ul>
6	<p><b>Languages and Grammars –</b></p> <ul style="list-style-type: none"> <li>a. Find the language generated by given grammar</li> <li>b. Check if a given string belongs or not to a given language/grammar</li> <li>c. Operations on languages</li> <li>d. Identify the type of grammar</li> </ul>
7	<p><b>Finite State Machines –</b></p> <ul style="list-style-type: none"> <li>a. Check if a given string is accepted or rejected by FSM without output</li> <li>b. Find the output for a FSM with output</li> <li>c. Describe a machine (diagram/table)</li> </ul>
8	<p><b>Regular Expression and Regular Language –</b></p> <ul style="list-style-type: none"> <li>a. Describe the regular expressions represented by given language</li> <li>b. Describe the language represented by given regular expression</li> </ul>
9	<p><b>Graphs –</b></p> <ul style="list-style-type: none"> <li>a. Types of graph</li> <li>b. Properties of graph</li> <li>c. Representation of graph</li> <li>d. Graph Isomorphism</li> <li>e. Connectivity in graphs – path, trail, walk</li> <li>f. Euler and Hamilton graphs</li> <li>g. Planar graphs</li> <li>h. Graph coloring and chromatic number</li> </ul>
10	<p><b>Trees –</b></p> <ul style="list-style-type: none"> <li>a. Tree terminologies</li> <li>b. Types of tree</li> <li>c. Properties of tree</li> <li>d. Representation of tree</li> <li>e. Expression tree</li> <li>f. Binary Search tree</li> <li>g. Tree traversal</li> </ul>

Course Code	Course Title	Credits	Lectures /Week
USCS106	Descriptive Statistics	2	3
<p><b>About the Course:</b>  This course is designed to provide learners with an understanding of the data and to develop an understanding of the quantitative techniques from Statistics. It also provides the knowledge of different statistical tools used for primary statistical analysis of data.</p>			
<p><b>Course Objectives:</b></p> <ol style="list-style-type: none"> <li>To develop the learners ability to deal with different types of data.</li> <li>To enable the use of different measures of central tendency and dispersion wherever relevant.</li> <li>To make learner aware about the techniques to check the Skewness and Kurtosis of data.</li> <li>To make learner enable to find the correlation between different variables and further apply the regression analysis to find the exact relation between them.</li> <li>To develop ability to analyze statistical data through R software.</li> </ol>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, learners would be able to</p> <ol style="list-style-type: none"> <li>Organize, manage and present data.</li> <li>Analyze Statistical data using measures of central tendency and dispersion.</li> <li>Analyze Statistical data using basics techniques of R.</li> <li>Study the relationship between variables using techniques of correlation and regression.</li> </ol>			
Unit	Topics	No of Lectures	
I	<p><b>Data Types and Data Presentation: Data types:</b> Attribute, Variable, Discrete and Continuous variable, Univariate and Bivariate distribution. Types of Characteristics, Different types of scales: nominal, ordinal, interval and ratio.</p> <p><b>Data presentation:</b> Frequency distribution, Histogram, Ogive curves.</p> <p><b>Introduction to R:</b> Data input, Arithmetic Operators, Vector Operations, Matrix Operations, Data Frames, Built-in Functions. Frequency Distribution, Grouped Frequency Distribution, Diagrams and Graphs, Summary statistics for raw data and grouped frequency distribution.</p> <p><b>Measures of Central tendency:</b> Concept of average/central tendency, characteristics of good measure of central tendency. Arithmetic Mean (A.M.), Median, Mode - Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, merits and demerits. Combined arithmetic mean. Partition Values: Quartiles, Deciles and Percentiles - examples for ungrouped and grouped data</p>	15	
II	<p><b>Measures dispersion:</b> Concept of dispersion, Absolute and Relative</p>	15	



	<p>measure of dispersion, characteristics of good measure of dispersion. Range, Semi-interquartile range, Quartile deviation, Standard deviation - Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, merits and demerits. Combined standard deviation, Variance. Coefficient of range, Coefficient of quartile deviation and Coefficient of variation (C.V.)</p> <p><b>Moments:</b> Concept of Moments, Raw moments, Central moments, Relation between raw and central moments.</p> <p><b>Measures of Skewness and Kurtosis:</b> Concept of Skewness and Kurtosis, measures based on moments, quartiles.</p>	
<b>III</b>	<p><b>Correlation:</b> Concept of correlation, Types and interpretation, Measure of Correlation: Scatter diagram and interpretation; Karl Pearson's coefficient of correlation (r): Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, properties; Spearman's rank correlation coefficient: Definition, examples of with and without repetition. Concept of Multiple correlation.</p> <p><b>Regression:</b> Concept of dependent (response) and independent (predictor) variables, concept of regression, Types and prediction, difference between correlation and regression, Relation between correlation and regression. Linear Regression - Definition, examples using least square method and regression coefficient, coefficient of determination, properties. Concept of Multiple regression and Logistic regression.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>Goon, A. M., Gupta, M. K. and Dasgupta, B. (1983). Fundamentals of Statistics, Vol. 1, Sixth Revised Edition, The World Press Pvt. Ltd., Calcutta.</li> <li>Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>Sarma, K. V. S. (2001). Statistics Made it Simple: Do it yourself on PC. Prentce Hall of India, NewDelhi.</li> <li>Agarwal, B. L. (2003). Programmed Statistics, Second Edition, New Age International Publishers, NewDelhi.</li> <li>Purohit, S. G., Gore S. D., Deshmukh S. R. (2008). Statistics Using R, Narosa Publishing House, NewDelhi.</li> <li>Schaum's Outline Of Theory And Problems Of Beginning Statistics, Larry J. Stephens, Schaum's Outline Series Mcgraw-Hill</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
-------------	--------------	---------	----------------

USCSP106	Descriptive Statistics – Practical	1	3
<b>Problem solving and implementation using R programming</b>			
1	<b>Basics of R-</b> <ol style="list-style-type: none"> <li>Data input, Arithmetic Operators</li> <li>Vector Operations, Matrix Operations</li> <li>Data Frames, Built-in Functions</li> <li>Frequency Distribution, Grouped Frequency Distribution</li> <li>Diagrams and Graphs</li> </ol>		
2	<b>Frequency distribution and data presentation-</b> <ol style="list-style-type: none"> <li>Frequency Distribution (Univariate data/ Bivariate data)</li> <li>Diagrams</li> <li>Graphs</li> </ol>		
3	<b>Measures of Central Tendency-</b> <ol style="list-style-type: none"> <li>Arithmetic Mean</li> <li>Median</li> <li>Mode</li> <li>Partition Values</li> </ol>		
4	<b>Measures dispersion-</b> <ol style="list-style-type: none"> <li>Range and Coefficient of range</li> <li>Quartile deviation and Coefficient of quartile deviation</li> <li>Standard deviation, Variance and Coefficient of variation (C.V.)</li> </ol>		
5	<b>Moments-</b> <ol style="list-style-type: none"> <li>Raw moments</li> <li>Central moments</li> </ol>		
6	<b>Measures of Skewness -</b> <ol style="list-style-type: none"> <li>Karl Pearson’s measure of Skewness</li> <li>Bowley’s measure of Skewness</li> <li>Moment coefficient of Skewness</li> </ol>		
7	<b>Measures of Kurtosis-</b> <ol style="list-style-type: none"> <li>Moment coefficient of Kurtosis (Absolute measure)</li> <li>Moment coefficient of Kurtosis (Relative measure)</li> </ol>		
8	<b>Correlation-</b> <ol style="list-style-type: none"> <li>Karl Pearson’s correlation coefficient</li> <li>Spearman’s Rank correlation</li> </ol>		
9	<b>Regression-</b> <ol style="list-style-type: none"> <li>Method of least squares</li> <li>Using regression coefficients</li> <li>Properties of regression lines &amp; regression coefficients</li> </ol>		
10	<b>Summary Statistics using R-</b> <ol style="list-style-type: none"> <li>Summary statistics for raw data</li> <li>Summary statistics for grouped frequency distribution</li> <li>Simple Correlation &amp; Regression using R</li> </ol>		
Course Code	Course Title	Credits	Lectures /Week

USCS107	Soft Skills	2	3
<b>About the Course:</b>			
To help learners develop their soft skills and develop their personality along with technical skills. Focus on various communication enhancement along with academic and professional ethics.			
<b>Course Objectives:</b>			
<ul style="list-style-type: none"> <li>• Understand the significance and essence of a wide range of soft skills.</li> <li>• Learn how to apply soft skills in a wide range of routine social and professional settings</li> <li>• Learn how to employ soft skills to improve interpersonal relationships</li> <li>• Learn how to employ soft skills to enhance employability and ensure workplace and career success</li> </ul>			
<b>Learning Outcomes:</b>			
<ul style="list-style-type: none"> <li>• Learners will be able to understand the importance and types soft skills</li> <li>• Learners will develop skills for Academic and Professional Presentations.</li> <li>• Learners will able to understand Leadership Qualities and Ethics.</li> <li>• Ability to understand the importance of stress management in their academic &amp; professional life.</li> </ul>			
<b>Unit</b>	<b>Topics</b>	<b>No of Lectures</b>	
<b>I</b>	<p><b>Introduction to Soft Skills</b>  <b>Soft Skills: An Introduction</b> – Definition and Significance of Soft Skills; Process, Importance and Measurement of Soft Skill Development.</p> <p><b>Personality Development:</b> Knowing Yourself, Positive Thinking, Johari’s Window, Physical Fitness</p> <p><b>Emotional Intelligence:</b> Meaning and Definition, Need for Emotional Intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence, Competencies of Emotional Intelligence, Skills to Develop Emotional Intelligence</p> <p><b>Positivity and Motivation:</b> Developing Positive Thinking and Attitude; Driving out Negativity; Meaning and Theories of Motivation; Enhancing Motivation Levels</p> <p><b>Etiquette and Mannerism:</b> Introduction, Professional Etiquette, Technology Etiquette</p> <p><b>Ethical Values:</b> Ethics and Society, Theories of Ethics, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics</p>	<b>15</b>	
<b>II</b>	<p><b>Basic Skills in Communication:</b>  <b>Components of effective communication:</b> Communication process and handling them, Composing effective messages, Non – Verbal</p>	<b>15</b>	

	<p>Communication: its importance and nuances: Facial Expression, Posture, Gesture, Eye contact, appearance (dress code).</p> <p><b>Communication Skills:</b> Spoken English, Phonetics, Accent, Intonation</p> <p><b>Employment Communication:</b> Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter</p> <p><b>Job Interviews:</b> Introduction, Importance of Resume, Definition of Interview, Background Information, Types of Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, Changes in the Interview Process, FAQ During Interviews</p> <p><b>Group Discussion:</b> Introduction, Ambience/Seating Arrangement for Group Discussion, Importance of Group Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits</p>	
<p style="text-align: center;"><b>III</b></p>	<p><b>Academic and Professional Skills:</b></p> <p><b>Professional Presentation:</b> Nature of Oral Presentation, planning a Presentation, Preparing the Presentation, Delivering the Presentation</p> <p><b>Creativity at Workplace:</b> Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method.</p> <p><b>Capacity Building: Learn, Unlearn and Relearn:</b> Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building</p> <p><b>Leadership and Team Building:</b> Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams.</p> <p><b>Decision Making and Negotiation:</b> Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts</p> <p><b>Stress and Time Management:</b> Stress, Sources of Stress, Ways to Cope with Stress</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Managing Soft Skills for Personality Development – edited by B.N.Ghosh, McGraw Hill India, 2017.</li> <li>2. Soft Skills: An Integrated Approach to Maximize Personality, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Personality Development and Soft Skills, Barun K. Mitra, Oxford Press</li> <li>2. Business Communication, ShaliniKalia, Shailja Agrawal, Wiley India</li> <li>3. Cornerstone: Developing Soft Skills, Sherfield, Pearson India</li> </ol>		

## Semester II

Course Code	Course Title	Credits	Lectures /Week
USCS201	Design & Analysis of Algorithms	2	3
<p><b>About the Course:</b>  The course covers the concepts of - (i) calculating complexity of algorithms, (ii) the essential operations like searching, sorting, selection, pattern matching &amp; recursion, and (iii) various algorithmic strategies like greedy, divide-n-conquer, dynamic programming, backtracking and implementations of all these on basic data structures like array, list and stack.</p>			
<p><b>Course Objectives:</b>  The objectives of this course are:</p> <ul style="list-style-type: none"> <li>To make students understand the basic principles of algorithm design</li> <li>To give idea to students about the theoretical background of the basic data structures</li> <li>To familiarize the students with fundamental problem-solving strategies like searching, sorting, selection, recursion and help them to evaluate efficiencies of various algorithms.</li> <li>To teach students the important algorithm design paradigms and how they can be used to solve various real world problems.</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Students should be able to understand and evaluate efficiency of the programs that they write based on performance of the algorithms used.</li> <li>Students should be able to appreciate the use of various data structures as per need</li> <li>To select, decide and apply appropriate design principle by understanding the requirements of any real life problems</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to algorithms</b> - What is algorithm, analysis of algorithm, Types of complexity, Running time analysis, How to Compare Algorithms, Rate of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-<math>\Omega</math> Notation, Theta-<math>\Theta</math> Notation, Asymptotic Analysis, Performance characteristics of algorithms, Estimating running time / number of steps of executions on paper, Idea of Computability</p> <p><b>Introduction to Data Structures</b> - What is data structure, types, Introduction to Array(1-d &amp; 2-d), Stack and List data structures, operations on these data structures, advantages disadvantages and applications of these data structures like solving linear equations, Polynomial Representation, Infix-to-Postfix conversion</p>	15	
II	<p><b>Recursion</b> - What is recursion, Recursion vs Iteration, recursion applications like Factorial of a number, Fibonacci series &amp; their comparative analysis with respect to iterative version, Tower of hanoi</p>	15	

	<p>problem</p> <p><b>Basic Sorting Techniques</b> - Bubble, Selection and Insertion Sort &amp; their comparative analysis</p> <p><b>Searching Techniques</b> - Linear Search and its types, Binary Search and their comparative analysis</p> <p><b>Selection Techniques</b> - Selection by Sorting, Partition-based Selection Algorithm, Finding the Kth Smallest Elements in Sorted Order &amp; their comparative analysis</p> <p><b>String Algorithms</b> - Pattern matching in strings, Brute Force Method &amp; their comparative analysis</p>	
<b>III</b>	<p><b>Algorithm Design Techniques</b> - Introduction to various types of classifications/design criteria and design techniques</p> <p><b>Greedy Technique</b> - Concept, Advantages &amp; Disadvantages, Applications, Implementation using problems like - file merging problem</p> <p><b>Divide-n-Conquer</b> - Concept, Advantages &amp; Disadvantages, Applications, Implementation using problems like - merge sort, Strassen's Matrix Multiplication</p> <p><b>Dynamic Programming</b> - Concept, Advantages &amp; Disadvantages, Applications, Implementation using problems like - Fibonacci series, Factorial of a number, Longest Common subsequence</p> <p><b>Backtracking Programming</b> - Concept, Advantages &amp; Disadvantages, Applications, Implementation using problems like N-Queen Problem</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. “Data Structure and Algorithm Using Python”, Rance D. Necaise, Wiley India Edition, 2016.</li> <li>2. “Data Structures and Algorithms Made Easy”, NarasimhaKarumanchi, CareerMonk Publications, 2016.</li> <li>3. “Introduction to Algorithms”, Thomas H. Cormen, 3rd Edition, PHI.</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. “Introduction to the Design and Analysis of Algorithms”, Anany Levitin, Pearson, 3rd Edition, 2011.</li> <li>2. “Design and Analysis of Algorithms”, S. Sridhar, Oxford University Press, 2014.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP201	Design & Analysis of Algorithms – Practical	1	3

1	Programs on 1-d arrays like - sum of elements of array, searching an element in array, finding minimum and maximum element in array, count the number of even and odd numbers in array. For all such programs, also find the time complexity, compare if there are multiple methods
2	Programs on 2-d arrays like row-sum, column-sum, sum of diagonal elements, addition of two matrices , multiplication of two matrices. For all such programs, also find the time complexity, compare if there are multiple methods
3	Program to create a list-based stack and perform various stack operations.
4	Program to perform linear search and binary search on list of elements. Compare the algorithms by calculating time required in milliseconds using readymade libraries.
5	Programs to sort elements of list by using various algorithms like bubble, selection sort, and insertion sort. Compare the efficiency of algorithms.
6	Programs to select the N <sup>th</sup> Max/Min element in a list by using various algorithms. Compare the efficiency of algorithms.
7	Programs to find a pattern in a given string - general way and brute force technique. Compare the efficiency of algorithms.
8	Programs on recursion like factorial, fibonacci, tower of hanoi. Compare algorithms to find factorial/fibonacci using iterative and recursive approaches.
9	Program to implement file merging, coin change problems using Greedy Algorithm and to understand time complexity.
10	Program to implement merge sort, Straseen's Matrix Multiplication using D-n-C Algorithm and to understand time complexity.
11	Program to implement fibonacci series, Longest Common Subsequence using dynamic programming and to understand time complexity. Compare it with the general recursive algorithm.
12	Program to implement N-Queen Problem, Binary String generation using Backtracking Strategy and to understand time complexity.

Course Code	Course Title	Credits	Lectures /Week
USCS202	Advanced Python Programming	2	3
<p><b>About the Course:</b>  This course aims to explore and enable learners to master the skills of advanced topics in Python Programming. It helps learners develop advanced skills such as working with databases, matching patterns, implementing threads and exception handling and GUI in Python. It also highlights and why Python is a useful scripting language for all developers.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To learn how to design object-oriented programs with Python classes.</li> <li>To learn about reading, writing and implementing other operation on files in Python.</li> <li>To implement threading concept and multithreading on Python</li> <li>To design GUI Programs and implement database interaction using Python.</li> <li>To know about use of regular expression and handling exceptions for writing robust python programs.</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Ability to implement OOP concepts in Python including Inheritance and Polymorphism</li> <li>Ability to work with files and perform operations on it using Python.</li> <li>Ability to implement regular expression and concept of threads for developing efficient program</li> <li>Ability to implement exception handling in Python applications for error handling.</li> <li>Knowledge of working with databases, designing GUI in Python and implement networking in Python</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Working with files:</b> Files, opening and closing a file, working with text files containing strings, knowing whether a file exists or not, working with binary files, the 'with' statement, the seek() and tell() methods, random accessing of binary files, zipping and unzipping files, working with directories, running other programs from python program</p> <p><b>Regular expressions:</b> What is a regular expression?, sequence characters in regular expressions, quantifiers in regular expressions, special characters in regular expressions, using regular expression on files, retrieving information from an html file,</p> <p><b>Threads in python:</b> Difference between process and thread, types of threads, benefits of threads, creating threads, single tasking and multitasking, thread synchronization, deadlock in threads, daemon threads</p> <p><b>Date and time in python:</b> Date and time now, combining date and time,</p>	15	



	formatting dates and times, finding durations using “time delta”, comparing two dates, sorting dates, stopping execution temporarily, knowing the time taken by a program, calendar module	
<b>II</b>	<p><b>Database in python:</b> Using SQL with python, retrieving rows from a table, inserting rows into a table, deleting rows from a table, updating rows in a table, creating database tables through python, Exception handling in databases.</p> <p><b>Exceptions in python:</b> Errors in a python program, compile &amp; run-time errors, logical error, exceptions-exception handling, types of exceptions, the except block, the assert statement, user-defined exceptions, logging the exceptions</p> <p><b>Networking:</b> Protocols,server-client architecture, tcp/ip and udp communication</p> <p><b>Graphical user interface:</b> Creating a GUI in python, Widget classes, Working with Fonts and Colours, working with Frames, Layout manager, Event handling</p>	<b>15</b>
<b>III</b>	<p><b>OOPs in python:</b> Features of Object Oriented Programming system (oops)-classes and objects, encapsulation, abstraction, inheritance, polymorphism, constructors and destructors</p> <p><b>Classes and objects:</b> Creating a class, the self-variable, types of variables, namespaces, types of methods, instance methods, class methods, static methods, passing members of one class to another class, inner classes</p> <p><b>Inheritance and polymorphism:</b> Inheritance in python, types of inheritance- single inheritance, multilevel inheritance, hierarchical inheritance, multiple inheritance, constructors in inheritance, overriding super class constructors and methods, the super() method, method resolution order (mro), polymorphism, duck typing, operator overloading, method overloading, method overriding,</p> <p><b>Abstract classes and interfaces:</b> Abstract class, abstract method, interfaces in python, abstract classes vs. Interfaces</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 3rd Edition, 2018</li> <li>2. Programming through Python, M. T Savaliya, R. K. Maurya, G M Magar, Revised Edition, Sybgen Learning India, 2020</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Advanced Python Programming, Dr. Gabriele Lanaro, Quan Nguyen, SakisKasampalis, Packt Publishing, 2019</li> <li>2. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018</li> <li>3. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018</li> <li>4. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017</li> </ol>		

5. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP202</b>	<b>Advanced Python Programming – Practical</b>	<b>1</b>	<b>3</b>
1	Write a program to Python program to implement various file operations.		
2	Write a program to Python program to demonstrate use of regular expression for suitable application.		
3	Write a Program to demonstrate concept of threading and multitasking in Python.		
4	Write a Python Program to work with databases in Python to perform operations such as <ul style="list-style-type: none"> <li>a. Connecting to database</li> <li>b. Creating and dropping tables</li> <li>c. Inserting and updating into tables.</li> </ul>		
5	Write a Python Program to demonstrate different types of exception handing.		
6	Write a GUI Program in Python to design application that demonstrates <ul style="list-style-type: none"> <li>a. Different fonts and colors</li> <li>b. Different Layout Managers</li> <li>c. Event Handling</li> </ul>		
7	Write Python Program to create application which uses date and time in Python.		
8	Write a Python program to create server-client and exchange basic information		
9	Write a program to Python program to implement concepts of OOP such as <ul style="list-style-type: none"> <li>a. Types of Methods</li> <li>b. Inheritance</li> <li>c. Polymorphism</li> </ul>		
10	Write a program to Python program to implement concepts of OOP such as <ul style="list-style-type: none"> <li>a. Abstract methods and classes</li> <li>b. Interfaces</li> </ul>		

Course Code	Course Title	Credits	Lectures /Week
USCS203	Introduction to OOPs using C++	2	3
<p><b>About the Course:</b>  The course aims to introduce a new programming paradigm called Object Oriented Programming. This will be covered using C++ programming language. C++ is a versatile programming language, which supports a variety of programming styles, including procedural, object-oriented, and functional programming. This makes C++ powerful as well as flexible. It can be used to develop software such as operating systems, databases, and compilers.</p>			
<p><b>Course Objectives:</b></p>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Work with numeric, character and textual data and arrays.</li> <li>• Understand the importance of OOP approach over procedural language.</li> <li>• Understand how to model classes and relationships using UML.</li> <li>• Apply the concepts of OOPS like encapsulation, inheritance and polymorphism.</li> <li>• Handle basic file operations.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Programming Concepts:</b> Object oriented programming paradigm, basic concepts of object oriented programming, benefits of object oriented programming, object oriented languages, applications of object oriented programming.</p> <p>Tokens-keywords, identifiers, constants-integer, real, character and string constants, backslash constants, features of C++ and its basic structure, simple C++ program without class, compiling and running C++ program.</p> <p><b>Data Types, Data Input Output and Operators:</b> Basic data types, variables, rules for naming variables, programming constants, the type cast operator, implicit and explicit type casting, cout and cin statements, operators, precedence of operators.</p> <p><b>Decision Making, Loops, Arrays and Strings:</b> Conditional statements-if, if...else, switch loops- while, do...while, for, types of arrays and string and string manipulations</p> <p><b>Unified Modeling Language (UML):</b> Introduction to UML &amp; class diagrams.</p> <p><b>Classes, Abstraction &amp; Encapsulation:</b> Classes and objects, Dot Operator, data members, member functions, passing data to functions, scope and visibility of variables in function.</p>	15	
II	<p><b>Constructors and Destructors:</b> Default constructor, parameterized</p>	15	

	<p>constructor, copy constructor, private constructor, destructors.</p> <p><b>Working with objects:</b> Accessor - mutator methods, static data and static function, access specifiers, array of objects.</p> <p><b>Polymorphism</b> - Binding-static binding &amp; overloading, constructor overloading function overloading, operator overloading, overloading unary and binary operators.</p> <p><b>Modelling Relationships in Class Diagrams:</b> Association, Aggregation-Composition and examples covering these principles</p>	
<p style="text-align: center;"><b>III</b></p>	<p><b>Inheritance:</b> Defining base class and its derived class, access specifiers, types of inheritance-single, multiple, hierarchical, multilevel, hybrid inheritance, friend function and friend class, constructors in derived classes.</p> <p><b>Modelling Relationships:</b> Generalization-Specialization and examples covering these principles</p> <p><b>Run time Polymorphism</b> - Dynamic Binding, Function overriding, virtual function, pure virtual function, virtual base class, abstract class.</p> <p><b>Pointers:</b> Introduction to pointers, * and &amp; operators, assigning addresses to pointer variables, accessing values using pointers, pointers to objects &amp; this pointer, pointers to derived classes</p> <p><b>File Handling:</b> File Stream classes, opening and closing file-file opening modes, text file handling, binary file handling.</p> <p><b>Applying OOP to solve real life applications:</b> To cover case studies like library management, order management etc. to design classes covering all relationships</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Object Oriented Programming with C++, Balagurusamy E., 8th Edition, McGraw Hill Education India.</li> <li>2. UML &amp; C++: A Practical Guide to Object Oriented Development, Lee/Tepfenhart, Pearson Education, 2<sup>nd</sup> Edition 2015</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Mastering C++ by Venugopal, Publisher: McGraw-Hill Education, 2017</li> <li>2. Let Us C++ by Kanetkar Yashwant, Publisher: BPB Publications, 2020</li> <li>3. Object Oriented Analysis and Design by Timothy Budd TMH, 2001</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP203</b>	<b>Introduction to OOPs using C++ – Practical</b>	<b>1</b>	<b>3</b>
1	Program to demonstrate use of data members & member functions.		
2	Programs based on branching and looping statements using classes.		
3	Program to demonstrate one and two dimensional arrays using classes		
4	Program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.		
5	Programs to demonstrate various types of constructors and destructors.		
6	Programs to demonstrate use of public, protected & private scope specifiers.		
7	Programs to demonstrate single and multilevel inheritance		
8	Programs to demonstrate multiple inheritance and hierarchical inheritance		
9	Programs to demonstrate inheritance and derived class constructors		
10	Programs to demonstrate friend function, inline function, this pointer		
11	Programs to demonstrate function overloading and overriding.		
12	Programs to demonstrate use of pointers		
13	Programs to demonstrate text and binary file handling		

Course Code	Course Title	Credits	Lectures /Week
USCS204	Database Systems	2	3
<p><b>About the Course:</b>  The course introduces the core principles and techniques required in the design and implementation of database systems. It includes ER Model, Normalization, Relational Model, and Relational Algebra. It also provides students with theoretical knowledge and practical skills of creating and manipulating data with an interactive query language (MySQL). It also provide student knowledge and importance of data protection.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To make students aware fundamentals of database system.</li> <li>• To give idea how ERD components helpful in database design and implementation.</li> <li>• To experience the students working with database using MySQL.</li> <li>• To familiarize the student with normalization, database protection and different DCL Statements.</li> <li>• To make students aware about importance of protecting data from unauthorized users.</li> <li>• To make students aware of granting and revoking rights of data manipulation.</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• To appreciate the importance of database design.</li> <li>• Analyze database requirements and determine the entities involved in the system and their relationship to one another.</li> <li>• Write simple queries to MySQL related to String, Maths and Date Functions.</li> <li>• Create tables and insert/update/delete data, and query data in a relational DBMS using MySQL commands.</li> <li>• Understand the normalization and its role in the database design process.</li> <li>• Handle data permissions.</li> <li>• Create indexes and understands the role of Indexes in optimization search.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to DBMS</b> – Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture</p> <p><b>Data models</b> - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network)</p> <p><b>Entity Relationship Model</b> - Entities, attributes, entity sets, relations, relationship sets, Additional constraints (key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER (entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER)</p>	15	

	<p><b>ER to Table-</b> Entity to Table, Relationship to tables with and without key constraints.</p> <p><b>DDL Statements</b> - Creating Databases, Using Databases, datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables</p> <p><b>DML Statements</b> – Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause</p>	
<b>II</b>	<p><b>Relational data model</b>– Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint</p> <p><b>Relational Algebra</b> operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division)</p> <p><b>Functions</b> – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse)</p> <p><b>Joining Tables</b> – inner join, outer join (left outer, right outer, full outer)</p> <p><b>Subqueries</b> – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries</p>	<b>15</b>
<b>III</b>	<p><b>Schema refinement and Normal forms:</b> Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition.</p> <p><b>Database Protection:</b> Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control, Backing Up and Restoring databases</p> <p><b>Views</b> (creating, altering dropping, renaming and manipulating views)</p> <p><b>DCL Statements</b> (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges), Transaction control commands – Commit, Rollback</p> <p><b>Index Structures of Files:</b> Introduction, Primary index, Clustering Index, Multilevel indexes</p>	<b>15</b>

**Textbooks:**

1. “Fundamentals of Database System”, ElmasriRamez, NavatheShamkant, Pearson Education, Seventh edition, 2017
2. “Database Management Systems”, Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014
3. “Murach's MySQL”, Joel Murach, 3rd Edition, 3rd Edition, 2019

**Additional References:**

1. “Database System Concepts”, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill, 2017
2. “MySQL: The Complete Reference”, Vikram Vaswani, McGraw Hill, 2017
3. “Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease”, Ashwin Pajankar, BPB Publications, 2020

Course Code	Course Title	Credits	Lectures /Week
USCSP204	Database Systems – Practical	1	3
1.	Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)		
2.	Perform the following: <ul style="list-style-type: none"> <li>• Viewing all databases</li> <li>• Creating a Database</li> <li>• Viewing all Tables in a Database</li> <li>• Creating Tables (With and Without Constraints)</li> <li>• Inserting/Updating/Deleting Records in a Table</li> </ul>		
3.	Perform the following: <ul style="list-style-type: none"> <li>• Altering a Table</li> <li>• Dropping/Truncating/Renaming Tables</li> <li>• Backing up / Restoring a Database</li> </ul>		
4.	Perform the following: <ul style="list-style-type: none"> <li>• Simple Queries</li> <li>• Simple Queries with Aggregate functions</li> </ul>		
5.	Queries involving <ul style="list-style-type: none"> <li>• Date Functions</li> <li>• String Functions</li> <li>• Math Functions</li> </ul>		
6.	Join Queries <ul style="list-style-type: none"> <li>• Inner Join</li> <li>• Outer Join</li> </ul>		



7.	<p>Subqueries</p> <ul style="list-style-type: none"> <li>• With IN clause</li> <li>• With EXISTS clause</li> </ul>
8.	<p>Converting ER Model to Relational Model and apply Normalization on database. (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys and normalization up to 3<sup>rd</sup> Normal Form).</p>
9.	<p>Views</p> <ul style="list-style-type: none"> <li>• Creating Views (with and without check option)</li> <li>• Dropping views</li> <li>• Selecting from a view</li> </ul>
10.	<p>DCL statements</p> <ul style="list-style-type: none"> <li>• Granting and revoking permissions</li> <li>• Saving (Commit) and Undoing (rollback)</li> </ul>
11.	<p>Creating Indexes on data tables.</p>

Course Code	Course Title	Credits	Lectures /Week
USCS205	Calculus	2	3
<p><b>About the Course:</b> Calculus is a branch of mathematics that involves the study of rates of change. In Computer Science, Calculus is used in Machine Learning, Data Mining, Scientific Computing, Image Processing, and creating the graphics and physics engines for video games, including the 3D visuals for simulations.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>The primary objective of this course is to introduce the basic tools of Calculus which are helpful in understanding their applications to the real world problems.</li> <li>The course is designed to have a grasp of important concepts of Calculus in a scientific way.</li> <li>It covers topics from as basic as definition of functions to partial derivatives of functions in a gradual and logical way.</li> <li>The learner is expected to solve as many examples as possible to get complete clarity and understanding of the topics covered.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, learners would be able to:</p> <ul style="list-style-type: none"> <li>Develop mathematical skills and enhance thinking power of learners.</li> <li>Understand mathematical concepts like limit, continuity, derivative, integration of functions, partial derivatives.</li> <li>Appreciate real world applications which uses the learned concepts.</li> <li>Skill to formulate a problem through Mathematical modelling and simulation.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>DERIVATIVES AND ITS APPLICATIONS:</b> <b>Review of Basic Concepts:</b> Functions, limit of a function, continuity of a function, derivative function.</p> <p><b>Derivative In Graphing And Applications:</b> Increase, Decrease, Concavity, Relative Extreme; Graphing Polynomials, Rational Functions, Cusps and Vertical Tangents. Absolute Maxima and Minima, Applied Maximum and Minimum Problems, Newton's Method.</p>	15	
II	<p><b>INTEGRATION AND ITS APPLICATIONS:</b> <b>Integration:</b> An Overview of the Area Problem, Indefinite Integral, Definition of Area as a Limit; Sigma Notation, Definite Integral, Evaluating Definite Integrals by Substitution, Numerical Integration: Simpson's Rule.</p> <p><b>Applications of Integration:</b> Area between two curves, Length of a plane curve.</p> <p><b>Mathematical Modeling with Differential Equations:</b> Modeling with</p>	15	

	Differential Equations, Separation of Variables, Slope Fields, Euler's Method, First-Order Differential Equations and Applications.	
<b>III</b>	<p><b>PARTIAL DERIVATIVES AND ITS APPLICATIONS:</b></p> <p><b>Functions of Several Variables:</b> Functions of two or more variables, Limits and Continuity of functions of two or three variables.</p> <p><b>Partial Derivatives:</b> Partial Derivatives, Differentiability, Differentials, and Local Linearity, Chain Rule, Implicit Differentiation, Directional Derivatives and Gradients,</p> <p><b>Applications of Partial Derivatives:</b> Tangent Planes and Normal Vectors, Maxima and Minima of Functions of Two Variables.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>Calculus: Early transcendental (10th Edition): Howard Anton, Irl Bivens, Stephen Davis, John Wiley &amp; sons, 2012.</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>Calculus and analytic geometry (9th edition): George B Thomas, Ross L Finney, Addison Wesley, 1995</li> <li>Calculus: Early Transcendentals (8th Edition): James Stewart, Brooks Cole, 2015.</li> <li>Calculus (10th Edition): Ron Larson, Bruce H. Edwards, Cengage Learning, 2013.</li> <li>Thomas' Calculus (13th Edition): George B. Thomas, Maurice D. Weir, Joel R. Hass, Pearson, 2014.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP205	Calculus – Practical	1	3
1	<p><b>Review of Basic Concepts –</b></p> <ol style="list-style-type: none"> <li>Functions of one variable, its domain and range, Operations on functions</li> <li>Limits of functions of one variable</li> <li>Continuity of functions of one variable</li> <li>Derivatives of functions of one variable</li> </ol>		
2	<p><b>Applications of Derivatives I –</b></p> <ol style="list-style-type: none"> <li>Increasing and Decreasing functions</li> <li>Concavity and inflection points</li> <li>Relative Extrema</li> <li>Absolute Extrema</li> </ol>		
3	<p><b>Applications of Derivatives II –</b></p> <ol style="list-style-type: none"> <li>Analysis of polynomials</li> <li>Graphing rational functions</li> <li>Graphs With Vertical Tangents And Cusps</li> <li>Newton's method to find approximate solution of an equation</li> </ol>		

4	<b>Integration –</b> <ol style="list-style-type: none"> <li>Finding area using rectangle method and antiderivative method</li> <li>Indefinite and definite integrals</li> <li>Properties of integrals</li> <li>Numerical integration using Simpson’s rule.</li> </ol>
5	<b>Applications of Integration –</b> <ol style="list-style-type: none"> <li>Area between two curves</li> <li>Length of a plane curve</li> </ol>
6	<b>Differential Equations –</b> <ol style="list-style-type: none"> <li>Solution of a first order first degree differential equation using variable separable method</li> <li>Solution of a first order linear differential equation using integrating factor</li> <li>Numerical solution of first-order equations using Euler’s method</li> <li>Modeling using differential equation</li> </ol>
7	<b>Functions of Several Variables –</b> <ol style="list-style-type: none"> <li>Functions of two or more variables, its domain and range, Operations on functions, level curves</li> <li>Limits of functions of two or three variables</li> <li>Continuity of functions of two or three variables</li> </ol>
8	<b>Partial Derivatives I –</b> <ol style="list-style-type: none"> <li>Partial derivatives of functions, First and Second order partial derivatives, Mixed derivative theorem, Higher order partial derivatives</li> <li>Differential for functions of two or three variables</li> <li>Local linear approximation for functions of two or three variables</li> </ol>
9	<b>Partial Derivatives II –</b> <ol style="list-style-type: none"> <li>Chain rule for functions of two or three variables</li> <li>Implicit differentiation</li> <li>Directional derivatives and gradient</li> </ol>
10	<b>Applications of Partial Derivatives–</b> <ol style="list-style-type: none"> <li>Tangent Planes and Normal Vectors for functions of two or three variables</li> <li>Maxima and Minima of Functions of Two Variables</li> </ol>
NOTE	<b>Above Practicals can also to be implemented using SageMath/ Geogebra.</b>

Course Code	Course Title	Credits	Lectures /Week									
USCS206	Statistical Methods	2	3									
<p><b>About the Course:</b>  This course introduces the key concepts in probability, conditional probabilities and distribution theory, including probability laws, random variables, expectation and variance, functions of random variables and its probability distributions. Emphasis is placed on theoretical understanding combined with problem solving using various statistical inferential techniques.</p>												
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To make learner aware about basic probability axioms and rules and its application.</li> <li>To understand the concept of conditional probability and Independence of events.</li> <li>To make learner familiar with discrete and continuous random variables as well as standard discrete and continuous distributions.</li> <li>To learn computational skills to implement various statistical inferential approaches.</li> </ul>												
<p><b>Learning Outcomes:</b>  After successful completion of this course, learners would be able to</p> <ul style="list-style-type: none"> <li>Calculate probability, conditional probability and independence.</li> <li>Apply the given discrete and continuous distributions whenever necessary.</li> <li>Define null hypothesis, alternative hypothesis, level of significance, test statistic and p value.</li> <li>Perform Test of Hypothesis as well as calculate confidence interval for a population parameter for single sample and two sample cases.</li> <li>Apply non-parametric test whenever necessary.</li> <li>Conduct and interpret one-way and two-way ANOVA.</li> </ul>												
<table border="1"> <thead> <tr> <th>Unit</th> <th>Topics</th> <th>No of Lectures</th> </tr> </thead> <tbody> <tr> <td>I</td> <td> <p><b>Probability:</b> Random experiment, sample space, events types and operations of events, Probability definition: classical, axiomatic, Elementary Theorems of probability (without proof). Conditional probability, 'Bayes' theorem, independence, Examples on Probability.</p> <p><b>Random Variables:</b> Concept and definition of a discrete random variable and continuous random variable. Probability mass function, Probability density function and cumulative distribution function of discrete and continuous random variable, Properties of cumulative distribution function.</p> </td> <td>15</td> </tr> <tr> <td>II</td> <td> <p><b>Mathematical Expectation and Variance:</b> Expectation of a function, Variance and S.D of a random variable, properties.</p> <p><b>Standard Probability distributions:</b> Introduction, properties, examples and applications of each of the following distributions: Binomial distribution, Normal distribution, Chi-square distribution, t distribution, F distribution</p> </td> <td>15</td> </tr> </tbody> </table>				Unit	Topics	No of Lectures	I	<p><b>Probability:</b> Random experiment, sample space, events types and operations of events, Probability definition: classical, axiomatic, Elementary Theorems of probability (without proof). Conditional probability, 'Bayes' theorem, independence, Examples on Probability.</p> <p><b>Random Variables:</b> Concept and definition of a discrete random variable and continuous random variable. Probability mass function, Probability density function and cumulative distribution function of discrete and continuous random variable, Properties of cumulative distribution function.</p>	15	II	<p><b>Mathematical Expectation and Variance:</b> Expectation of a function, Variance and S.D of a random variable, properties.</p> <p><b>Standard Probability distributions:</b> Introduction, properties, examples and applications of each of the following distributions: Binomial distribution, Normal distribution, Chi-square distribution, t distribution, F distribution</p>	15
Unit	Topics	No of Lectures										
I	<p><b>Probability:</b> Random experiment, sample space, events types and operations of events, Probability definition: classical, axiomatic, Elementary Theorems of probability (without proof). Conditional probability, 'Bayes' theorem, independence, Examples on Probability.</p> <p><b>Random Variables:</b> Concept and definition of a discrete random variable and continuous random variable. Probability mass function, Probability density function and cumulative distribution function of discrete and continuous random variable, Properties of cumulative distribution function.</p>	15										
II	<p><b>Mathematical Expectation and Variance:</b> Expectation of a function, Variance and S.D of a random variable, properties.</p> <p><b>Standard Probability distributions:</b> Introduction, properties, examples and applications of each of the following distributions: Binomial distribution, Normal distribution, Chi-square distribution, t distribution, F distribution</p>	15										

<b>III</b>	<p><b>Hypothesis testing:</b> One sided, Two sided hypothesis, critical region, p-value, tests based on t, Normal and F, confidence intervals.</p> <p><b>Analysis of Variance:</b> One-way, two-way analysis of variance.</p> <p><b>Non-parametric tests:</b> Need of non-parametric tests, Sign test, Wilcoxon's signed rank test, run test, Kruskal-Walis tests, Chi square test.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi</li> <li>Goon, A. M., Gupta, M. K. and Dasgupta, B. (1983). Fundamentals of Statistics, Vol. 1, Sixth Revised Edition, The World Press Pvt. Ltd., Calcutta.</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>Mood, A. M. and Graybill, F. A. and Boes D.C. (1974). Introduction to the Theory of Statistics, Ed. 3, McGraw Hill Book Company.</li> <li>Hoel P. G. (1971). Introduction to Mathematical Statistics, John Wiley and Sons, New York.</li> <li>Hogg, R.V. and Craig R.G. (1989). Introduction to Mathematical Statistics, Ed. MacMillan Publishing Co., New York.</li> <li>Walpole R. E., Myers R. H. and Myers S. L. (1985), Probability and Statistics for Engineers and Scientists</li> <li>Agarwal, B. L. (2003). Programmed Statistics, Second Edition, New Age International Publishers, New Delhi.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP206	Statistical Methods – Practical	1	3
1	<p><b>Probability-</b></p> <ol style="list-style-type: none"> <li>Examples based on Probability definition: classical, axiomatic</li> <li>Examples based on elementary Theorems of probability</li> </ol>		
2	<p><b>Conditional probability and independence-</b></p> <ol style="list-style-type: none"> <li>Examples based on Conditional probability</li> <li>Examples based on 'Bayes' theorem</li> <li>Examples based on independence</li> </ol>		
3	<p><b>Discrete random variable-</b></p> <ol style="list-style-type: none"> <li>Probability distribution of discrete random variable</li> <li>Probability mass function</li> </ol>		
4	<p><b>Continuous random variable-</b></p> <ol style="list-style-type: none"> <li>Probability distribution of continuous random variable</li> <li>Probability density function</li> </ol>		

5	<b>Mathematical Expectation and Variance-</b> <ol style="list-style-type: none"> <li>Mean of discrete and continuous Probability distribution</li> <li>S.D. and variance of discrete and continuous Probability distribution</li> </ol>
6	<b>Standard probability distributions-</b> <ol style="list-style-type: none"> <li>Calculation of probability, mean and variance based on Binomial distribution</li> <li>Calculation of probability based on Normal distribution</li> </ol>
7	<b>Large Sample tests based on Normal (Z) -</b> <ol style="list-style-type: none"> <li>Test of significance for proportion (Single proportion <math>H_0: P = P_0</math>)</li> <li>Test of significance for difference between two proportions (Double proportion <math>H_0: P_1 = P_2</math>)</li> <li>Test of significance for mean (Single mean <math>H_0: \mu = \mu_0</math>)</li> <li>Test of significance for difference between two means. (Double mean <math>H_0: \mu_1 = \mu_2</math>)</li> </ol>
8	<b>Small sample tests based on t and F-</b> <ol style="list-style-type: none"> <li>t-test for significance of single mean, population variance being unknown (Single mean <math>H_0: \mu = \mu_0</math>)</li> <li>t-test for significance of the difference between two sample means (Independent samples)</li> <li>t-test for significance of the difference between two sample means (Related samples)</li> <li>F-Test to Compare Two Variances</li> </ol>
9	<b>Analysis of variance -</b> <ol style="list-style-type: none"> <li>Perform One-way ANOVA</li> <li>Perform Two-way ANOVA</li> </ol>
10	<b>Non-parametric tests-</b> <ol style="list-style-type: none"> <li>Sign test and Wilcoxon Sign rank test</li> <li>Run test</li> <li>Kruskal-Wallis (H) test</li> <li>Chi-square test</li> </ol>
<b>Note: Practical no. 6, 7, 8, 9 can also to be implemented using R programming.</b>	

Course Code	Course Title	Credits	Lectures /Week
USCS207	E-Commerce & Digital Marketing	2	3
<p><b>About the Course:</b>  This course introduces the fundamental concepts of e-commerce, its types, the various legal and ethical issues of e-commerce and different e-commerce applications. The course also aims to introduce basic principles and types of digital marketing and web and Google analytics</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To understand increasing significance of E-Commerce and its applications in Business and Various Sectors</li> <li>To provide an insight on Digital Marketing activities on various Social Media platforms and its emerging significance in Business</li> <li>To understand Latest Trends and Practices in E-Commerce and Digital Marketing, along with its Challenges and Opportunities for an Organization</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Understand the core concepts of E-Commerce.</li> <li>Understand the various online payment techniques</li> <li>Understand the core concepts of digital marketing and the role of digital marketing in business.</li> <li>Apply digital marketing strategies to increase sales and growth of business</li> <li>Apply digital marketing through different channels and platforms</li> <li>Understand the significance of Web Analytics and Google Analytics and apply the same.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to E-Commerce and E- Business:</b> Definition and competing in the digital economy, Impact of E-Commerce on Business Models, Factors Driving e-commerce and e-Business Models, Economics and social impact of e-Business, opportunities and Challenges, e-Commerce vs m-Commerce, Different e-Commerce Models (B2B, B2C, C2B, C2C, B2E), e-Commerce Applications: e-Trading, e-Learning, e-Shopping, Virtual Reality &amp; Consumer Experience, Legal and Ethical issues in e-Commerce.</p> <p><b>Overview of Electronic Payment systems:</b> Types of Electronic payment schemes (Credit cards, Debit cards, Smartcards, Internet banking), E-checks, E-Cash Concepts and applications of EDI and Limitation</p> <p><b>Introduction &amp; origin of Digital Marketing:</b> Traditional v/s Digital Marketing. Digital Marketing Strategy, The P-O-E-M Framework, Segmenting &amp; Customizing Messages, The Digital landscape, Digital Advertising Market in India. Skills required in Digital Marketing. Digital Marketing Plan.</p>	15	
II	<p><b>Social Media Marketing:</b> Meaning, Purpose, types of social media websites, Social Media Engagement, Target audience, Facebook Marketing: Business through Facebook Marketing, Creating Advertising Campaigns,</p>	15	



	<p>Adverts, Facebook Marketing Tools, LinkedIn Marketing: Importance of LinkedIn Marketing, Framing LinkedIn Strategy, Lead Generation through LinkedIn, Content Strategy, Analytics and Targeting, Twitter Marketing: Framing content strategy, Twitter Advertising Campaigns, YouTube Marketing: Video optimization, Promoting on YouTube, Monetization, YouTube Analytics</p> <p><b>Email Marketing:</b> Types of Emails, Mailing List, Email Marketing tools, Email Deliverability &amp; Email Marketing automation</p> <p><b>Mobile Marketing:</b> Introduction, Mobile Usage, Mobile Advertising, Mobile Marketing Types, Mobile Marketing Features, Mobile Campaign Development, Mobile Advertising Analytics</p> <p><b>Content Marketing:</b> Introduction, Content marketing statistics, Types of Content, Types of Blog posts, Content Creation, Content optimization, Content Management &amp; Distribution, Content Marketing Strategy, Content creation tools and apps, Challenges of Content Marketing.</p>	
III	<p><b>Search Engine Optimization:</b> Meaning, Common SEO techniques, Understanding Search Engines, basics of Keyword search, Google rankings, Link Building, Steps to optimize website, On-page and off-page optimization</p> <p><b>Search Engine Marketing:</b> Introduction to SEM, Introduction to Ad Words - Google Ad Words, Ad Words fundamentals, Ad Placement, Ad Ranks, Creating Ad Campaigns, Campaign Report Generation, Display marketing, Buying Models: Cost per Click (CPC), Cost per Milli (CPM), Cost per Lead (CPL), Cost per Acquisition (CPA).</p> <p><b>Web Analytics:</b> Purpose, History, Goals &amp; objectives, Web Analytic tools &amp; Methods. Web Analytics Mistakes and Pitfalls.</p> <p><b>Google Analytics:</b> Basics of Google Analytics, Installing Google Analytics in website, Parameters of Google Analytics, Reporting and Analysis</p>	15
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. “E-Commerce Strategy, Technologies and Applications”, Whitley, David, Tata McGraw Hill, 2017</li> <li>2. Digital Marketing, Seema Gupta, McGraw Hill Education, 2<sup>nd</sup> Edition</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. E-Commerce by S. Pankaj, A.P.H. Publication, New Delhi</li> <li>2. Fundamentals of Digital Marketing, Punit Singh Bhatia, Pearson, 2<sup>nd</sup> Edition</li> <li>3. “Understanding Digital Marketing: MarketingStrategies for Engaging the Digital Generation”, Damian Ryan, Calvin Jone. Kogan Page, 4<sup>th</sup> Edition</li> </ol>		

## EvaluationScheme

### I. Internal Evaluation for Theory Courses – 25 Marks

#### (i) Mid-Term Class Test– 15Marks

- It should be conducted using any **learning management system** such as **Moodle**(Modularobject-orienteddynamiclearning environment)
- The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes**.

#### (ii) Assignment/ Case study/ Presentations– 10 Marks

- Assignment / Case Study Report / Presentation can be uploaded onany **learning management system**.

### II. External Examination for Theory Courses – 75 Marks

- Duration: **2.5 Hours**
- Theory question paper pattern:

<b>All questions are compulsory.</b>			
<b>Question</b>	<b>Based on</b>	<b>Options</b>	<b>Marks</b>
Q.1	Unit I	<i>Any 4 out of 6</i>	20
Q.2	Unit II	<i>Any 4 out of 6</i>	20
Q.3	Unit III	<i>Any 4 out of 6</i>	20
Q.4	Unit I,II and III	<i>Any 5 out of 6</i>	15

- All questions shall be compulsorywith internal choicewithin thequestions.
- Each Question maybe sub-divided into subquestions as a, b, c, d, etc.&the allocation ofMarks dependson theweightage ofthetopic.

### III. Practical Examination

- Each core subjectcarries50 Marks  
**40 marks + 05 marks (journal) + 05 marks (viva)**
- Duration: **2 Hours**for each practical course.
- Minimum **80% practical** from each core subjects are required to be completed.
- **Certified Journal is compulsory for appearing at the time of Practical Exam**
- The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

**UNIVERSITY OF MUMBAI**



**Syllabus for**  
**Program: Bachelor of Science**  
**Course: Computer Science**

*with effect from*  
**Academic Year 2022-2023**

## Preamble

The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skills sets demanded under new technological environment. It also endeavors to align the programme structure and course curriculum with student aspirations and corporate expectations. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present day context.

The Core Subjects offers to develop strong theoretical foundations in Computer Science to build computational thinking, analytical, and problem solving skills. Principles of Operating Systems course provides an overview of computer operating systems, their functionalities, processes, and computing resource management. Linear Algebra course covers concepts crucial to many areas of computer science, such as graphics, image processing, cryptography, machine learning, computer vision, optimization, graph algorithms, quantum computation, computational biology, information retrieval and web search. Data Structures course provides an understanding of different types of data structures and how to use them per the requirements of a given application. Advanced Database Concepts course touches the touches security, recovery, and transaction aspects of database. Theory of Computation course helps to develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas. Computer Networks course include topics such as application layer protocols, Internet protocols, network interfaces, local and wide area networks, wireless networks, bridging and routing, among other current topics. Software Engineering course embodies an engineering approach to the development of software. It discusses the nature of software and software projects, software development models, software process maturity, project planning, management, and estimations along with topics on software testing and quality assurance. The course on IoT Technologies will definitely open future area as Embedded Engineer, involvement in IoT projects, Robotics and many more.

Skill Enhancement courses such as Java based Application Development, Web Technologies, Android Application Development and Advanced Application Development cater to present day needs of web and mobile based platforms and applications. These courses aims to produce skilled graduates with a creative mind-set who can recognize a computational problem either in IT industry or society, and develop effective solutions.

The General Elective courses offers the students the option to explore disciplines of interest beyond the choices they make in Core and Discipline Specific Elective papers. The course on Creative Content Writing prepare students to comprehend, refine, and enhance their writing abilities and enter the industry with enhanced skill and substantial competence. The course on Green Technologies emphasizes the use of principles and practices of green services and regulatory standards for addressing the carbon issues and related concerns. The Research Methodology instills basic research skills for students who wish to pursue a research or an academic career. Management & Entrepreneurship course aims to focus on giving students the business management and innovation skills required to succeed in a startup.

We sincerely believe that any student taking this programme will get very strong foundation and exposure to basics, advanced and emerging trends of the subject.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents, we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to University Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly helped designing certain specialized courses and the syllabus as a whole.

# S.Y.B.Sc. Computer Science Syllabus

## Choice Based Credit System (CBCS)

*with effect from*

**Academic year 2022-2023**

<b>Semester – III</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
USCS301	Core Subject	Principles of Operating Systems	2	3
USCSP301	Core Subject Practical	Principles of Operating Systems – Practical	1	3
USCS302	Core Subject	Linear Algebra	2	3
USCSP302	Core Subject Practical	Linear Algebra – Practical	1	3
USCS303	Core Subject	Data Structures	2	3
USCSP303	Core Subject Practical	Data Structures – Practical	1	3
USCS304	Core Subject	Advanced Database Concepts	2	3
USCSP304	Core Subject Practical	Advanced Database Concepts – Practical	1	3
USCS305	Skill Enhancement Course (SEC)	Java based Application Development	2	3
USCSP305	Skill Enhancement Course (SEC) Practical	Java based Application Development – Practical	1	3
USCS306	Skill Enhancement Course (SEC)	Web Technologies	2	3
USCSP306	Skill Enhancement Course (SEC) Practical	Web Technologies – Practical	1	3
USCS3071	Generic Elective	Creative Content Writing	2	3
USCS3072	Generic Elective	Green Technologies	2	3

*\* Any one Generic Elective has to be selected by the student.*

# S.Y.B.Sc. Computer Science Syllabus

## Choice Based Credit System (CBCS)

*with effect from*

**Academic year 2022-2023**

<b>Semester – IV</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
USCS401	Core Subject	Theory of Computation	2	3
USCSP401	Core Subject Practical	Theory of Computation – Practical	1	3
USCS402	Core Subject	Computer Networks	2	3
USCSP402	Core Subject Practical	Computer Networks – Practical	1	3
USCS403	Core Subject	Software Engineering	2	3
USCSP403	Core Subject Practical	Software Engineering – Practical	1	3
USCS404	Core Subject	IoT Technologies	2	3
USCSP404	Core Subject Practical	IoT Technologies – Practical	1	3
USCS405	Skill Enhancement Course (SEC)	Android Application Development	2	3
USCSP405	Skill Enhancement Course (SEC) Practical	Android Application Development – Practical	1	3
USCS406	Skill Enhancement Course (SEC)	Advanced Application Development	2	3
USCSP406	Skill Enhancement Course (SEC) Practical	Advanced Application Development – Practical	1	3
USCS4071	Generic Elective*	Research Methodology	2	3
USCS4072	Generic Elective*	Management & Entrepreneurship	2	3

*\* Any one Generic Elective has to be selected by the student.*

## Semester III

Course Code	Course Title	Credits	Lectures /Week
USCS301	Principles of Operating Systems	2	3
<p><b>About the Course:</b> The purpose of this course is to provide an overview of computer operating systems, their functionalities, processes, and computing resource management. In particular, the course will cover processes and threads, mutual exclusion, CPU scheduling, deadlock, memory management, and file systems.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To learn basic concepts and structure of operating systems</li> <li>• To learn about process and synchronization in operating system level</li> <li>• To learn CPU scheduling algorithms</li> <li>• To learn Memory and File system management</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Work with any type of operating system</li> <li>• Handle threads, processes, process synchronization</li> <li>• Implement CPU scheduling algorithms</li> <li>• Understand the background role of memory management</li> <li>• Design file system.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Operating-Systems:</b> Definition of Operating System, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments</p> <p><b>Operating-System Structures:</b> Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure</p> <p><b>Processes:</b> Process Concept, Process Scheduling, Operations on Processes, Inter process Communication</p> <p><b>Threads:</b> Overview, Multicore Programming, Multithreading Models</p>	15	
II	<p><b>Process Synchronization:</b> General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors</p>	15	

	<p><b>CPU Scheduling:</b> Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling</p> <p><b>Deadlocks:</b> System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock</p>	
III	<p><b>Main Memory:</b> Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table</p> <p><b>Virtual Memory:</b> Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing</p> <p><b>Mass-Storage Structure:</b> Overview, Disk Structure, Disk Scheduling, Disk Management</p> <p><b>File-System Interface:</b> File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing</p> <p><b>File-System Implementation:</b> File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management</p>	15
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 2021</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill, 2017</li> <li>2. Naresh Chauhan, Principles of Operating Systems, Oxford Press, 2014</li> <li>3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016</li> </ol>		



Course Code	Course Title	Credits	Lectures /Week
USCSP301	Principles of Operating Systems – Practical	1	3
1	Process Communication: <ol style="list-style-type: none"> <li>Write a program to give a solution to the producer–consumer problem using shared memory.</li> <li>Write a program to give a solution to the producer–consumer problem using message passing.</li> </ol>		
2	Threads: <ol style="list-style-type: none"> <li>Write a program to work with a single thread.</li> <li>Write a program to work with multi threads.</li> <li>The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, ... Formally, it can be expressed as: <math>fib_0 = 0</math>, <math>fib_1 = 1</math>, <math>fib_n = fib_{n-1} + fib_{n-2}</math>. Write a multithreaded program that generates the Fibonacci sequence.</li> </ol>		
3	Synchronization: <ol style="list-style-type: none"> <li>Write a program to give a solution to the Bounded buffer problem.</li> <li>Write a program to give a solution to the readers–writers problem.</li> </ol>		
4	Write a program that implements FCFS scheduling algorithm.		
5	Write a program that implements (with no preemption) scheduling algorithm.		
6	Write a program that implements RR scheduling algorithm.		
7	Write a program that implements the banker’s algorithm		
8	Write a program that implements the FIFO page-replacement algorithm.		
9	Write a program that implements the LRU page-replacement algorithm.		
10	Write a program to design a File System.		

Course Code	Course Title	Credits	Lectures /Week
USCS302	Linear Algebra	2	3
<p><b>About the Course:</b>            Linear algebra, a branch of mathematics, provides concepts that are crucial to many areas of computer science, such as graphics, image processing, cryptography, machine learning, computer vision, optimization, graph algorithms, quantum computation, computational biology, information retrieval and web search. The course covers topics such as fields, vectors, matrices, eigenvalues and eigenvectors</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To offer the learner the relevant Linear Algebra concepts through Computer Science applications.</li> <li>• To interpret existence and analyze the solution set of a system of linear equations.</li> <li>• To formulate, solve, apply, and interpret properties of linear systems.</li> <li>• To learn about the concept of linear independence of vectors over a field, and the dimension of a vector space.</li> <li>• To interpret basic concepts of linear transformations, dimension, matrix representation of a linear transformation, and the change of coordinate matrix.</li> </ul>			
<p><b>Learning Outcomes:</b>            After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Appreciate the relevance and applications of Linear Algebra in the field of Computer Science.</li> <li>• Understand the concepts through program implementation.</li> <li>• Instill a computational thinking while learning linear algebra.</li> <li>• Express clear understanding of the concept of a solution to a system of equations.</li> <li>• Find eigenvalues and corresponding eigenvectors for a square matrix.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Field:</b> Introduction to complex numbers, complex numbers in Python, abstracting over fields, Playing with GF (2).</p> <p><b>Vectors:</b> Vectors are functions, Vector addition, Scalar-vector multiplication, combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations, Support Vector Machine – Introduction, Mechanism.</p> <p><b>The Vector Space:</b> Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise</p>	15	
II	<p><b>Matrix:</b> Matrices as vectors, Column space and row space, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product, From function inverse to matrix inverse</p>	15	

	<p><b>Basis:</b> Coordinate systems, two greedy algorithms for finding a set of generators, Linear dependence, Basis, Unique representation, Change of basis, first look, Computational problems involving finding a basis</p> <p><b>Dimension:</b> Dimension and rank, Direct sum, Dimension and linear functions, The annihilator</p> <p><b>Gaussian elimination:</b> Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination.</p>	
<b>III</b>	<p><b>Inner Product:</b> The inner product for vectors over the reals, Orthogonality.</p> <p><b>Orthogonalization:</b> Projection orthogonal to multiple vectors, projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, orthogonal complement.</p> <p><b>Eigenvalues and Eigenvectors:</b> Characteristic Polynomials of degree 2 and 3, Eigenvalues and eigenvectors, Properties of eigenvalues and eigenvectors, Cayley–Hamilton Theorem, Minimal Polynomial. Coordinate representation in terms of eigenvectors, The Internet worm, Markov Chains, Google Page Rank algorithm.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Coding the Matrix Linear Algebra through Applications to Computer Science, First Edition, Philip N. Klein, Newtonian Press 2013</li> <li>2. Schaum's Outline of Linear Algebra, Sixth Edition by Seymour Lipschutz, Marc Lipson, McGraw Hill 2017</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Linear Algebra and Probability for Computer Science Applications, First Edition, Ernest Davis, A K Peters/CRC Press, 2012.</li> <li>2. Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition, 2007</li> <li>3. Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition, 2002</li> <li>4. Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2008.</li> <li>5. Computer Networking With Internet Protocols and Technology, William Stallings, Pearson Education India, 2013.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP302	Linear Algebra – Practical	1	3
1	Write a program which demonstrates the following: <ul style="list-style-type: none"> <li>• Addition of two complex numbers</li> <li>• Displaying the conjugate of a complex number</li> <li>• Plotting a set of complex numbers</li> <li>• Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number <math>a = 1/2</math>, <math>a = 1/3</math>, <math>a = 2</math> etc.</li> </ul>		
2	Write a program to do the following: <ul style="list-style-type: none"> <li>• Enter a vector <math>u</math> as a <math>n</math>-list</li> <li>• Enter another vector <math>v</math> as a <math>n</math>-list</li> <li>• Find the vector <math>au + bv</math> for different values of <math>a</math> and <math>b</math></li> <li>• Find the dot product of <math>u</math> and <math>v</math></li> </ul>		
3	Vector Applications: Classify given data using support vector machines (SVM)		
4	Basic Matrix Operations: <ul style="list-style-type: none"> <li>• Matrix Addition, Subtraction, Multiplication</li> <li>• Check if matrix is invertible.</li> <li>• If yes then find Inverse</li> </ul>		
5	Write a program to convert a matrix into its row echelon form. (Order 2). Write a program to find rank of a matrix.		
6	Basic Matrix Application – I Representation of Image in Matrix Format and Image Transformations		
7	Basic Matrix Application – II Perform Image addition, multiplication and subtraction		
8	Write a program to do the following: <ul style="list-style-type: none"> <li>• Enter a vector <math>b</math> and find the projection of <math>b</math> orthogonal to a given vector <math>u</math>.</li> <li>• Find the projection of <math>b</math> orthogonal to a set of given vectors</li> </ul>		
9	Write a program to calculate eigenvalue and eigenvector (Order 2 and 3)		
10	Implement Google’s Page rank algorithm.		

Course Code	Course Title	Credits	Lectures /Week
USCS303	Data Structures	2	3
<p><b>About the Course:</b> The course focuses to give an understanding of different types of data structures that can be used to store data in memory, how to create-manipulate them and to use them in the best possible manner as per the requirements of the application.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To introduce data abstraction and data representation in memory</li> <li>• To describe, design and use of elementary data structures such as stack, queue, linked list, tree and graph</li> <li>• How and why different data structures are used for different types of problems.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to-</p> <ul style="list-style-type: none"> <li>• Create different types of data structures.</li> <li>• Understand which data structure to be used based on the type of the problem.</li> <li>• Apply combined knowledge of algorithms and data structures to write highly effective programs in various domains.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Abstract Data Type:</b> Different Data Types, different types of data structures &amp; their classifications, Introduction to ADT, Creating user-specific ADT</p> <p><b>Linked Structures:</b> ADT for linked list, Advantages &amp; Disadvantages, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, applications of linked list like polynomial equation</p> <p><b>Stacks:</b> Stack ADT for Stack, Advantages &amp; Disadvantages, Applications of stack like balanced delimiter, prefix to postfix notation</p> <p><b>Queues:</b> Queue ADT, Advantages &amp; Disadvantages, linked representations. Circular Queue operations, Dequeues, applications of queue like job scheduling queues</p>	15	
II	<p><b>Doubly Linked list:</b> ADT of doubly linked list, Advantages &amp; Disadvantages, Insertion and deletion of nodes at various positions</p> <p><b>Trees:</b> ADT for Tree Structure. Advantages &amp; disadvantages, Binary Tree-Properties, Implementation and Traversals, Binary Search Tree, Balanced BST, Threaded Binary Trees, AVL Trees, Applications of Tree like Huffman Coding,</p>		

	<b>Priority Queues &amp; Heaps:</b> Priority Queue, Priority Queue ADT, Advantages and Disadvantages, Applications, Heaps, types of heaps, Heapifying the element,	
<b>III</b>	<b>Graph:</b> Introduction, Graph ADT, Advantages and Disadvantages, Graph Representation using adjacency matrix and adjacency list, Graph operations like insertion and deletion of nodes, Graph Traversals using BFS & DFS, Applications of Graphs like shortest path algorithms, <b>Hashing:</b> Hash Table ADT, Advantages & Disadvantages, Concept of hashing, hash table, hash functions, collision, collision avoidance techniques, Applications of hashing	
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Introduction to Algorithm, Thomas H Cormen, PHI</li> <li>2. Data Structures And Algorithms Made Easy, Narasimha Karumanchi, 2021</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran Ellis Horowitz, Universities Press, 2018</li> <li>2. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley, 2016</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP303</b>	<b>Data Structures – Practical</b>	<b>1</b>	<b>3</b>
1	Write a program to implement Abstract Data Types (ADT)		
2	Write a program to implement Singly Linked list with insertion, deletion, traversal operations		
3	Write a program to implement Doubly Linked list with insertion, deletion, traversal operations		
4	Write a program to implement Stack with insertion, deletion, traversal operations		
5	Write a program to implement Queue with insertion, deletion, traversal operations		
6	Write a program to implement Priority Queue with insertion, deletion, traversal operations		
7	Write a program to implement Binary Tree with insertion, deletion, traversal operations		
8	Write a program to implement Huffman Coding		
9	Write a program to implement Graph with insertion, deletion, traversal operations		
10	Write a program to implement Travelling Salesman Problem		
11	Write a program to create basic Hash Table for insertion, deletion, traversal operations(assume that there are no collisions)		
12	Write a program to create hash table to handle collisions using overflow chaining		

Course Code	Course Title	Credits	Lectures /Week
USCS304	Advanced Database Concepts	2	3
<p><b>About the Course:</b> This course deals with the basic understanding of programming in database. It touches security, recovery, and transaction aspects of database. The course will increase the confidence among the learner while dealing with database.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage.</li> <li>To develop understanding of Transaction management and crash recovery.</li> <li>To develop concepts of programming concepts of database.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Master concepts of stored procedure, functions, cursors and triggers and its use.</li> <li>Learn about using PL/SQL for data management.</li> <li>Use efficiently Collections and records.</li> <li>Understand concepts and implementations of transaction management and crash recovery.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Overview of PL/SQL:</b> Advantages of PL/SQL, Main Features of PL/SQL, Architecture of PL/SQL</p> <p><b>Fundamentals of PL/SQL:</b> Character Sets, Lexical Units, Declarations, References to Identifiers, Scope and Visibility of Identifiers, Assigning Values to Variables, Expressions, Error-Reporting Functions, Data Types.</p> <p><b>Control Statements:</b> Conditional Selection Statements, LOOP Statements, Sequential Control Statements, GOTO, and NULL Statements.</p> <p><b>Sequences:</b> creating sequences, referencing, altering, and dropping a sequence.</p> <p><b>Stored Procedures and Functions:</b> Procedures: Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. Functions: Calling function and recursion function.</p>	15	
II	<p><b>Collections and Records:</b> Associative Arrays, Varrays (Variable-Size Arrays), Nested Tables, Collection Constructors, Assigning Values to Collection Variables, Multidimensional Collections, Collection</p>	15	



	<p>Comparisons, Collection Methods, Collection Types Defined in Package Specifications, Record Variables, Assigning Values to Record Variables.</p> <p><b>Error Handling:</b> Compile-Time Warnings, Overview of Exception Handling, Internally Defined Exceptions, Predefined Exceptions, User-Defined Exceptions, Redeclared Predefined Exceptions, Raising Exceptions Explicitly, Exception Propagation, Unhandled Exceptions.</p> <p><b>Cursors:</b> Overview of Cursor, Types of cursors, Invalid cursor Exception.</p> <p><b>Static and Dynamic SQL:</b> Static SQL: Description of Static SQL, Cursors Overview, Processing Query Result Sets, Cursor Variables, CURSOR Expressions, Transaction Processing and Control, Autonomous Transactions. Dynamic SQL: Native Dynamic SQL, DBMS_SQL Package, SQL Injection.</p>	
<b>III</b>	<p><b>Triggers:</b> Overview of Triggers, implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting, and modifying triggers, and enforcing data integrity through triggers.</p> <p><b>Packages:</b> Overview of a Package. Need of Packages, Package Specification, Package Body, Package Instantiation and Initialization.</p> <p><b>Transaction Management:</b> ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem, Inconsistent Read Problem, Read-Write Locks, Deadlocks Handling, Two Phase Locking protocol.</p> <p><b>Crash Recovery:</b> ARIES algorithm. The log-based recovery, recovery related structures like transaction and dirty page table, Write-ahead log protocol, check points, recovery from a system crash, Redo and Undo phases</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Mastering PL/SQL Through Illustrations: From Learning Fundamentals to Developing Efficient PL/SQL Blocks, Dr. B. Chandra, BPB Publication, 2020</li> <li>2. Oracle Pl/Sql Training Guide., Training guide, BPB Publications, 2016</li> <li>3. Raghu Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill,3rd Edition, 2014</li> <li>4. Abraham Silberschatz, Henry F. Korth,S. Sudarshan , Database System Concepts, 6th Edition 2019</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Ivan Bayross, “SQL, PL/SQL -The Programming language of Oracle”, B.P.B. Publications 2009</li> <li>2. Ramez Elmasri &amp; Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, 2008</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP304</b>	<b>Advanced Database Concepts – Practical</b>	<b>1</b>	<b>3</b>
1	Writing PL/SQL Blocks with basic programming constructs by including following: a. Sequential Statements b. unconstrained loop		
2	Sequences: a. Creating simple Sequences with clauses like START WITH, INCREMENT BY, MAXVALUE, MINVALUE, CYCLE   NOCYCLE, CACHE   NOCACHE, ORDER   NOORDER. b. Creating and using Sequences for tables.		
3	Writing PL/SQL Blocks with basic programming constructs by including following: a. If...then...Else, IF...ELSIF...ELSE... END IF b. Case statement		
4	Writing PL/SQL Blocks with basic programming constructs for following Iterative Structure: a. While-loop Statements b. For-loop Statements.		
5	Writing PL/SQL Blocks with basic programming constructs by including a GoTO to jump out of a loop and NULL as a statement inside IF.		
6	Writing Procedures in PL/SQL Block a. Create an empty procedure, replace a procedure and call procedure b. Create a stored procedure and call it c. Define procedure to insert data d. A forward declaration of procedure		
7	Writing Functions in PL/SQL Block. a. Define and call a function b. Define and use function in select clause, c. Call function in dbms_output.put_line d. Recursive function e. Count Employee from a function and return value back f. Call function and store the return value to a variable		
8	Creating and working with Insert/Update/Delete Trigger using Before/After clause.		
9	Write an Implicit and explicit cursor to complete the task.		
10	Create packages and use it in SQL block to complete the task.		
11	Write a SQL block to handle exception by writing: a. Predefined Exceptions, b. User-Defined Exceptions, c. Redeclared Predefined Exceptions,		
12	Create nested tables and work with nested tables.		

Course Code	Course Title	Credits	Lectures /Week
USCS305	Java based Application Development	2	3
<p><b>About the Course:</b> The objective of this course is to teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and explore advanced topic of Java programming for solving problems.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To provide insight into java based applications using OOP concepts.</li> <li>To provide understanding of developing GUI based desktop applications in java.</li> <li>To provide knowledge of web based applications through servlet and jsp.</li> <li>To provide understanding and implementation of basic JSON</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Design basic application in java using Graphical User Interface.</li> <li>The learner will be able to develop applications using swings</li> <li>The learner will be able to develop web based applications using servlet and jsp</li> <li>The learner will be able to connect databases with java through</li> <li>The learner will be able to perform programs using JSON objects</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> History, Features of Java, Java Development Kit, Java Application Programming Interface, Java Virtual Machine Java Program Structure, Java Tokens.</p> <p><b>OOPS:</b> Introduction, Class, Object, Static Keywords, Constructors, this keyword, Inheritance, Inner class, Anonymous Inner class, super keyword, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces</p> <p><b>Packages:</b> Introduction to predefined packages, User Defined Packages, Access specifiers</p> <p><b>Exception Handling:</b> Introduction, Pre-Defined Exceptions, try-catch-finally, throws, throw, User Defined Exceptions</p> <p><b>Multithreading:</b> Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, wait() notify() notify all() methods</p>	15	
II	<p><b>Collection Framework:</b> Introduction, java.util Package interfaces, List, Set, Map, List interface &amp; its classes, Set interface &amp; its classes, Map interface &amp; its classes.</p>	15	

	<p><b>Introduction to JFC and Swing-</b> Features of the Java Foundation Classes, Swing API Components, JComponent Class, Windows, Dialog Boxes, and Panels, Labels, Buttons, Check Boxes, Menus, Toolbars, Implementing Action interface, Pane, JScrollPane, Desktop pane, Scrollbars, Lists and Combo Boxes, Text-Entry Components, Colors and File Choosers, Tables and Trees, Printing with 2D API and Java Print Service API.</p> <p><b>Event Handling:</b> Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes.</p> <p><b>JDBC:</b> Introduction, JDBC Architecture, JDBC Drivers, JDBC Connectivity Model, java.sql package, Using Statement, PreparedStatement, CallableStatement, ResultSet, Scrollable and Updatable ResultSet, Navigating and manipulating data, ResultSetMetaData, Managing Transactions in JDBC, JDBC Exception classes, BLOB &amp; CLOB</p>	
<p style="text-align: center;"><b>III</b></p>	<p><b>Servlets:</b> Introduction, Servlet Life Cycle, Types of Servlet, Servlet Configuration with Deployment Descriptor, Working with ServletContext and ServletConfig Object, Attributes in Servlet,, Response and Redirection using Request Dispatcher and using sendRedirect Method, Filter API, Manipulating Responses using Filter API, Session Tracking: using Cookies, HttpSession, Hidden Form Fields and URL Rewriting,Types of Servlet Event: ContextLevel and SessionLevel.</p> <p><b>Java Server Pages (JSP):</b> Introduction to JSP , Comparison with Servlet, JSP Architecture, JSP Life Cycle, JSP Scripting Elements, JSP Directives, JSP Action, JSP Implicit Objects, JSP Expression Language, JSP Standard Tag Libraries, JSP Custom Tag, JSP Session Management, JSP Exception Handling, JSP CRUD Applications</p> <p><b>JSON:</b> Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Herbert Schildt, Java The Complete Reference, Eleventh Edition, McGraw-Hill Education, 2020</li> <li>2. Bryan Basham, Kathy Sierra, Bert Bates, Head First Servlets and JSP, O’reilly (SPD), 2018</li> <li>3. Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR, 2004</li> <li>4. Ivan Bayross, Web Enabled Commercial Applications Development Using Java 2, BPB Publications</li> <li>5. Java XML and JSON: Document Processing for Java SE by Jeff Friesen January 2019, Apress</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. E. Balagurusamy, Programming with Java- A Primer, Tata McGraw-Hill Education India, 2014</li> <li>2. Programming in JAVA, 2nd Ed, Sachin Malhotra &amp; Saurabh Choudhary, Oxford Press, 2018</li> <li>3. Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD)</li> <li>4. Eric Jendrock, Jennifer Ball, D Carson and others, The Java EE 5 Tutorial, Pearson Education</li> <li>5. The Java Tutorials: <a href="http://docs.oracle.com/javase/tutorial/">http://docs.oracle.com/javase/tutorial/</a></li> <li>6. Java Parsing Collection XML JSON: Map List XML JSON Transform by Yang Hu, 2019</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP305</b>	<b>Java based Application Development – Practical</b>	<b>1</b>	<b>3</b>
1	<ul style="list-style-type: none"> <li>a. Write a program to create a class and implement the concepts of Constructor Overloading, Method Overloading, Static methods</li> <li>b. Write a program to implement the concept of Inheritance and Method Overriding</li> </ul>		
2	<ul style="list-style-type: none"> <li>a. Write a program to implement the concepts of Abstract classes and methods</li> <li>b. Write a program to implement the concept of interfaces</li> </ul>		
3	Write a program to define user defined exceptions and raise them as per the requirements		
4	<p>Write a program to demonstrate the methods of:</p> <ul style="list-style-type: none"> <li>a. List interface</li> <li>b. Set interface</li> <li>c. Map interface</li> </ul>		
5	Write a program using various swing components design Java application to accept a student's resume. (Design form)		
6	<ul style="list-style-type: none"> <li>a. Write a JDBC program that displays the data of a given table</li> <li>b. Write a JDBC program to return the data of a specified record from a given table</li> <li>c. Write a JDBC program to insert / update / delete records into a given table</li> </ul>		
7	<ul style="list-style-type: none"> <li>a. Construct a simple calculator using the JAVA Swings with minimum functionality.</li> <li>b. Construct a GUI using JAVA Swings to accept details of a record of a given table and submit it to the database using JDBC technology on the click of a button.</li> </ul>		
8	<ul style="list-style-type: none"> <li>a. Write a Servlet that accepts a User Name from a HTML form and stores it as a cookie. Write another Servlet that returns the value of this cookie and displays it.</li> <li>b. Write a Servlet that displays the names and values of the cookie stored on the client.</li> <li>c. Write a Servlet that accepts a User Name from a HTML form and stores it as a session variable. Write another Servlet that returns the value of this session variable and displays it.</li> </ul>		
9	<ul style="list-style-type: none"> <li>a. Write a registration Servlet that accepts the data for a given table and stores it in the database.</li> <li>b. Write a Servlet that displays all the records of a table.</li> </ul>		
10	<ul style="list-style-type: none"> <li>a. Write a JSP that accepts a User Name from a HTML form and stores it as a cookie. Write another JSP that returns the value of this cookie and displays it.</li> <li>b. Write a JSP that displays the names and values of the cookie stored on the client.</li> <li>c. Write a JSP that accepts a User Name from a HTML form and stores it as a session variable. Write another JSP that returns the value of this session variable and displays it.</li> </ul>		

11	<ol style="list-style-type: none"><li>a. Write a JSP code that accepts username and password from HTML file and validates the user from the database</li><li>b. Write a registration JSP that accept the data for a given table and stores it in the database.</li><li>c. Write a JSP that displays all the records of a table</li></ol>
12	. Write Java application to encoding and decoding JSON in Java.

Course Code	Course Title	Credits	Lectures /Week
USCS306	Web Technologies	2	3
<p><b>About the Course:</b> The course provides an insight into emerging technologies to design and develop state of the art web applications using client-side scripting, server-side scripting, and database connectivity</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To understand the concepts of Hyper Text Markup Language and Cascading Style Sheets.</li> <li>• To learn JavaScript for creating dynamic websites.</li> <li>• To learn various operations performed on data among web applications using XML</li> <li>• To learn Server-Side Programming using PHP</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Design valid, well-formed, scalable, and meaningful pages using emerging technologies.</li> <li>• Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites</li> <li>• Develop and implement client-side and server-side scripting language programs.</li> <li>• Develop and implement Database Driven Websites.</li> <li>• Design and apply XML to create a markup language for data and document centric applications.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>HTML5:</b> Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page</p> <p><b>CSS:</b> Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element.</p>	15	
II	<p><b>JavaScript:</b> Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript – Variables, Operators, Control Flow Statements, Popup Boxes, Functions – Defining and Invoking a Function, Defining Function arguments, defining a return Statement, Calling Functions with Timer, JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window, Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript</p> <p><b>XML:</b> Comparing XML with HTML, Advantages and Disadvantages of XML, Structure of an XML Document, XML Entity References, DTD,</p>	15	

	XSLT: XSLT Elements and Attributes - xsl:template, xsl:apply-templates, xsl:import, xsl:call-template, xsl:include, xsl:element, xsl:attribute, xsl:attribute-set, xsl:value-of	
<b>III</b>	<p><b>AJAX:</b> AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties and Methods, handling asynchronous requests using AJAX</p> <p><b>PHP:</b> Variables and Operators, Program Flow, Arrays, working with Files and Directories, working with Databases, Working with Cookies, Sessions and Headers</p> <p><b>Introduction to jQuery:</b> Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press, 2016</li> <li>2. Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India, 2018</li> <li>3. PHP: A Beginners Guide, Vikram Vaswani, TMH</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY, 2011</li> <li>2. Learning PHP, MySQL, JavaScript, CSS &amp; HTML5, Robin Nixon, O'Reilly, 2018</li> <li>3. PHP, MySQL, JavaScript &amp; HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley, 2018</li> </ol>		



Course Code	Course Title	Credits	Lectures /Week
USCSP306	Web Technologies – Practical	1	3
1	Design a webpage that makes use of <ul style="list-style-type: none"> <li>a. Document Structure Tags</li> <li>b. Various Text Formatting Tags</li> <li>c. List Tags</li> <li>d. Image and Image Maps</li> </ul>		
2	Design a webpage that makes use of <ul style="list-style-type: none"> <li>a. Table tags</li> <li>b. Form Tags (forms with various form elements)</li> <li>c. Navigation across multiple pages</li> <li>d. Embedded Multimedia elements</li> </ul>		
3	Design a webpage that make use of Cascading Style Sheets with <ul style="list-style-type: none"> <li>a. CSS properties to change the background of a Page</li> <li>b. CSS properties to change Fonts and Text Styles</li> <li>c. CSS properties for positioning an element</li> </ul>		
4	Write JavaScript code for <ul style="list-style-type: none"> <li>a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number</li> <li>b. Validating the various Form Elements</li> </ul>		
5	Write JavaScript code for <ul style="list-style-type: none"> <li>a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date</li> <li>b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document,</li> <li>c. Storing and Retrieving Cookies</li> </ul>		
6	Create a XML file with Internal / External DTD and display it using <ul style="list-style-type: none"> <li>a. CSS</li> <li>b. XSL</li> </ul>		
7	Design a webpage to handle asynchronous requests using AJAX on <ul style="list-style-type: none"> <li>a. Mouseover</li> <li>b. button click</li> </ul>		
8	Write PHP scripts for <ul style="list-style-type: none"> <li>a. Retrieving data from HTML forms</li> <li>b. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number</li> <li>c. Working with Arrays</li> <li>d. Working with Files (Reading / Writing)</li> </ul>		
9	Write PHP scripts for		

	<ul style="list-style-type: none"><li>a. Working with Databases (Storing Records / Retrieving Records and Display them)</li><li>b. Storing and Retrieving Cookies</li><li>c. Storing and Retrieving Sessions</li></ul>
10	Design a webpage with some jQuery animation effects.

Course Code	Course Title	Credits	Lectures /Week
USCS3071	Creative Content Writing	2	3
<p><b>About the Course:</b>            With the advent of the internet, content writing has become a very lucrative and promising career. The course is designed to equip students to comprehend, refine, and enhance their writing abilities so that they may become proficient web content developers. The course aims to prepare students to enter the industry with enhanced skill and substantial competence.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To introduce students to the concepts of content writing.</li> <li>• To connect them with various writing and editing styles and techniques.</li> <li>• To help them develop their creative abilities.</li> <li>• To improve the learners' employability</li> </ul>			
<p><b>Learning Outcomes:</b>            After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Understand the fundamentals of content creation for Blog, Website etc.</li> <li>• Acquire the ability to write and edit in a variety of styles and procedures</li> <li>• To develop the creative abilities.</li> <li>• To acquire essential language skills for editors.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Basics of Content writing:</b> Introduction to Content Writing, Learning Tone in Writing and Its Types, Comprehending style in writing and its Types, Common Grammatical Errors.</p> <p><b>Best Practices for Writing for the Web:</b> Making our story Elegant, Professional, Write with an Attitude, Keep Verbs Active, List Items, Chunk Information, Title and Subtitle, Organize for Your Audience.</p> <p><b>Things Marketers Write:</b> The Ideal Length for Blog Posts, Podcast, Facebook Posts, Tweets, and Other Marketing Content.</p>	15	
II	<p><b>Social Media Writing:</b> Writing for Twitter, writing with Hashtags, Writing Social Media with Humor, writing for Facebook, writing for LinkedIn, Writing Your LinkedIn Profile, writing for Email, Writing Landing Pages, Writing Headlines, writing a Home Page, Writing the About Us Page, Writing Better Blog Posts, Writing Annual Reports.</p> <p><b>Infographics:</b> Visual Communication- What Are Infographics?, The Science of Visualization, Creating Infographics- Purpose, The Art of Observation, Processing Your Ideas, Designing Your Infographics, Publishing Your Infographics.</p>	15	

<b>III</b>	<p><b>Content Tools:</b> Research and Knowledge Management Tools, Writing Tools, Productivity Tools, Editing Tools, A Few Great Style Guides, Non-Text Writing Tools, Blog Idea Generators, Google Authorship, Image Sources, Tools for Content Writing.</p> <p><b>Ethical and Legal aspects of content writing:</b> Learn Legal English, Learn Legal Vocabulary In Legal Writing, IPR Laws, and Copywriting, Plagiarism laws in Content Writing.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Content Writing Handbook, Author:Kounal Gupta, 2020, Henry Harvin.</li> <li>2. Feldar, Lynda. Writing for the Web: Creating Compelling Web Content Using Words, Pictures, and Sound. New Riders, CA, USA, 2011</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Everybody Writes: Your Go-To Guide to Creating Ridiculously Good Content Paperback Ann Handley Pan Macmillan India 2016</li> <li>2. The Power of Infographics: Using Pictures to Communicate and Connect With Your Audiences Paperback – 15 June 2012 Mark Smiciklas</li> <li>3. Law Relating to Intellectual Property Rights Book by V. K. Ahuja, 2017</li> </ol> <p><b>Web Resources:</b></p> <ol style="list-style-type: none"> <li>1. <a href="https://www.locationrebel.com/b2b-writing/">https://www.locationrebel.com/b2b-writing/</a></li> <li>2. <a href="https://www.mindler.com/blog/how-to-become-a-content-writer-in-india/">https://www.mindler.com/blog/how-to-become-a-content-writer-in-india/</a></li> <li>3. <a href="https://study.com/articles/What_is_a_Content_Writer.html">https://study.com/articles/What_is_a_Content_Writer.html</a></li> <li>4. <a href="https://www.mondaq.com/india/contracts-and-commercial-law/445620/legal-contractsagreements-drafting-and-legal-vetting">https://www.mondaq.com/india/contracts-and-commercial-law/445620/legal-contractsagreements-drafting-and-legal-vetting</a></li> <li>5. <a href="https://www.crazyegg.com/blog/copywriting/">https://www.crazyegg.com/blog/copywriting/</a></li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCS3072	Green Technologies	2	3
<p><b>About the Course:</b>  This course focuses on familiarizing learners with the need and relevance of Green Computing, Technology, and its practices for creating a sustainable work and production environment for the IT-enabled sector. The course emphasizes the use of principles and practices of green services and regulatory standards for addressing the carbon issues and related concerns.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Know about Green IT Fundamentals: Business, IT, and the Environment</li> <li>• Green IT Strategies and Significance of Green IT Strategies</li> <li>• Green Enterprise Architecture and Green Information Systems</li> <li>• Sociocultural Aspects of Green IT and Green Compliance</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Explain drivers and dimensions of change for Green Technology</li> <li>• Appreciate Virtualization; smart meters and optimization in achieving green IT</li> <li>• Gain knowledge about green assets, green processes, and green enterprise architecture</li> <li>• ISO 14001 and related standards for Audit for Green Compliance</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Green IT Fundamentals:</b> Information Technology and Environment, Business, Environment, and Green Enterprise Characteristics, Green Vision and Strategic Points, Green Value, Green IT Opportunity, Challenges of a Carbon Economy, Environmental Intelligence, Envisioning the Green Future</p> <p><b>Green IT Strategies:</b> Green strategic alignment, Green IT Drivers-Cost, Regulatory and Legal, Sociocultural and Political, Business ecosystem, New market opportunities, Green IT Business Dimensions, KPIs in Green Strategies</p> <p><b>Environmentally Responsible Business:</b> Developing ERBS, Policies, Practices, and Metrics, Mobility and Environment, Green It Metrics and Measurements, Green IT Readiness and CMM, Context Sensitivity and Automation in Green IT Measures</p> <p><b>Green Assets:</b> Introduction, Green Assets, Green IT Hardware, Green Data Centers and ICT Equipment, Server and Data Strategy</p>	15	
II	<p><b>Green Assets and emerging Trends:</b> Data Servers Optimization and Virtualization, Physical Data Server Organization and Cooling, Cloud Computing and Data Centers, Networking and Communications Infrastructure, End-User Devices, Smart Meters in Real-Time, Managing</p>	15	

	<p>Devices for Central Green Services, Devices and Organizational Boundaries for Measurements, Mobile Devices, and Sustainability</p> <p><b>Green Business Process Management:</b> Introduction, Green Reengineering, Green Process, Green BPM and standards, Green Business Analysis, Green Requirements Modelling, Green IT Governance, Green Business Process and Applications, QoS, Achieving green BPM, Green Mobile Business Process, Digital Library</p> <p><b>Green Enterprise Architecture:</b> Green IT and organizational Systems, Aspects of Green Solutions Architecture, Contents and Integration with Service-Oriented Architecture, Green Supply Chain Management, Green Portals in Green Enterprise Architecture, Environmental Intelligence</p>	
<p style="text-align: center;"><b>III</b></p>	<p><b>Green Information Systems(GIS): Design and Development Models:</b> Describing GIS, GIS Requirements</p> <p><b>Sociocultural Aspects of Green IT:</b> Green IT's Social Impact, Learning Organization, Green Social Stakeholders, Role-Based View of Green IT, Green User Practices, Attitude and Subjectivity in Green IT, Green IT Ethics and Code of Conduct, Privacy and Security of Green Information, Green Washing, Communications in Green Transformation Projects, Green HR and Changing Organizational Structures, Green-Collar Workers: Roles and Skill Sets, Green Virtual Communities</p> <p><b>Green Compliance: Protocols, Standards, and Audits:</b> Protocols and Standards, ISO 14000-2004 Standard, Various initiatives by stakeholders, Green Audits and types, Audit and use of Carbon emission management software</p> <p><b>Emerging Carbon Issues:</b> Technologies and Future: Future Carbon Landscape, Green ICT and Technology Trends, Cloud Computing, Nanotechnology, Quantum computing, Renewable energies, eco-design, Collaborative environmental intelligence</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Green IT Strategies and Applications Using Environmental Intelligence, Bhuvan Unhelkar, CRC Press, 2016</li> <li>2. Green Information and Communication Systems for a Sustainable Future, Rajshree Srivastava, Sandeep Kautish, Rajeev Tiwari. CRC Press, 2020</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Emerging Green Technologies, Matthew N. O. Sadiku, Taylor and Francis (CRC Press), 2022</li> <li>2. Sustainability Awareness and Green Information Technologies, Tomayess Issa, Springer, 2021</li> <li>3. Environmental Sustainability Role of Green Technologies, P. Thangavel, and G. Sridevi, Springer, 2016</li> </ol>		

## Semester IV

Course Code	Course Title	Credits	Lectures /Week
USCS401	Theory of Computation	2	3
<p><b>About the Course:</b> The course provides a comprehensive insight into theory of computation by understanding grammar, languages and other elements of modern language design. It also helps to develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To give an overview of the theoretical foundations of computer science from the perspective of formal languages</li> <li>To illustrate finite state machines to solve problems in computing</li> <li>To explain the hierarchy of problems arising in the computer sciences.</li> <li>To familiarize Regular grammars, context free grammar.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Understand Grammar and Languages</li> <li>Learn about Automata theory and its application in Language Design</li> <li>Learn about Turing Machines and Pushdown Automata</li> <li>Understand Linear Bound Automata and its applications</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Automata Theory:</b> Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and N DFA equivalence, Mealy and Moore Machines, Minimizing Automata.</p> <p><b>Formal Languages:</b> Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata</p>	15	
II	<p><b>Regular Sets and Regular Grammar:</b> Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar</p> <p><b>Context Free Languages:</b> Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG</p> <p><b>Pushdown Automata:</b> Definitions, Acceptance by PDA, PDA and CFG</p>	15	

<b>III</b>	<p><b>Linear Bound Automata:</b> The Linear Bound Automata Model, Linear Bound Automata and Languages.</p> <p><b>Turing Machines:</b> Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine,</p> <p><b>Undecidability:</b> The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI, 3rd Edition 2019</li> <li>2. Introduction to Computer Theory, Daniel Cohen, Wiley, 2nd Edition, 2007</li> <li>3. Introductory Theory of Computer Science, E.V. Krishnamurthy, Affiliated East-West Press, 2009</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Theory of Computation, Kavi Mahesh, Wiley India, 2018</li> <li>2. Elements of The Theory of Computation, Lewis, Papadimitriou, PHI, 2015</li> <li>3. Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education, 2010</li> <li>4. Introduction to Theory of Computation, Michel Sipser, Thomson</li> <li>5. Introduction to Automata Theory, Languages and Computation, John E. Hopcroft, Pearson Education, 2014</li> </ol>		



<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP401</b>	<b>Theory of Computation – Practical</b>	<b>1</b>	<b>3</b>
1	Write a program for tokenization of given input		
2	Write a program for generating regular expressions for regular grammar		
3	Write a program for generating derivation sequence / language for the given sequence of productions		
4	Design a Program for creating machine that accepts three consecutive one.		
5	Design a Program for creating machine that accepts the string always ending with 101.		
6	Design a program for accepting decimal number divisible by 2.		
7	Design a program for creating a machine which accepts string having equal no. of 1's and 0's.		
8	Design a program for creating a machine which count number of 1's and 0's in a given string.		
9	Design a PDA to accept WCWR where w is any string and WR is reverse of that string and C is a Special symbol.		
10	Design a Turing machine that's accepts the following language $an b n c n$ where $n > 0$		

Course Code	Course Title	Credits	Lectures /Week
USCS402	Computer Networks	2	3
<p><b>About the Course:</b>  This course introduces computer networks, with a special focus on the Internet architecture and protocols. The course includes topics such as network architectures, addressing, naming, forwarding, routing, communication reliability, the client-server model, web, email and other application layer protocols.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To Understand Basic Concepts of Networking.</li> <li>• To Understand Working of Network Layer Architecture.</li> <li>• To Learn Practical Implementation of Basic Routing Algorithms.</li> <li>• To Learn Different Networking Protocols.</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Learn basic networking concepts and layered architecture.</li> <li>• Understand the concepts of networking, which are important for them to be known as a 'networking professionals'.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> Networking standards and Administrations, networks, network types – LAN, MAN, WAN.</p> <p><b>Network Models:</b> The OSI model, TCP/IP protocol suite,</p> <p><b>Introduction to Physical layer:</b> Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance.</p> <p><b>Digital transmissions:</b> Digital-to-digital conversion, analog-to-digital conversion, transmission modes</p> <p><b>Analog transmissions:</b> digital-to-analog conversion, analog-to-analog conversion.</p> <p><b>Bandwidth Utilization – Multiplexing and Spectrum spreading:</b> Multiplexing, Spread Spectrum</p> <p><b>Transmission media:</b> Guided Media, Unguided Media</p> <p><b>Switching:</b> Introduction, Circuit Switched Network, Packet Switching.</p>	15	
II	<p><b>Introduction to Data Link Layer:</b> Link layer addressing, Data Link Layer Design Issues.</p> <p><b>Error detection and correction:</b> -Block coding, cyclic codes, checksum, forward error correction, error correcting codes, error detecting codes.</p>	15	

	<p><b>Data Link Control:</b> DLC services, data link layer protocols, HDLC, Point-to-point protocol.</p> <p><b>Media Access Control:</b> Random access, controlled access, channelization,</p> <p><b>Wired LANs – Ethernet:</b> Ethernet Protocol, standard Ethernet, fast Ethernet, gigabit Ethernet, 10 gigabit Ethernet</p> <p><b>Wired Network:</b> Telephone Network, Cable Network, SONET, ATM</p> <p><b>Wireless LANs:</b> Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks.</p> <p><b>Introduction to Network Layer:</b> Network layer services, packet switching, network layer performance, IPv4 addressing, forwarding of IP packets,</p> <p><b>Network Layer Protocols :</b> Internet Protocol, ICMPv4, Mobile IP</p>	
<b>III</b>	<p><b>Unicast Routing:</b> Introduction, routing algorithms, unicast routing protocols.</p> <p><b>Next generation IP:</b> IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6.</p> <p><b>Introduction to the Transport Layer:</b> Transport Layer Protocol, User Datagram Protocol, Transmission Control Protocol, SCTP.</p> <p><b>Introduction to Application Layer:</b> Client Server Programming, Iterative Programming.</p> <p><b>Standard Client-Server Protocols:</b> WWW, HTTP, FTP, Electronic Mail, TELNET, Secure Cell, DNS, SNMP</p> <p><b>Quality of Service:</b> Data Flow to improve QoS, Flow control to improve QoS, Integrated service (Intserv), Differentiated Service(Diffserv).</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2018.</li> <li>2. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2018.</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Computer Network, Bhushan Trivedi, Oxford University Press, 2016</li> <li>2. Data and Computer Communication, William Stallings, PHI, 2017</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP402</b>	<b>Computer Networks – Practical</b>	<b>1</b>	<b>3</b>
1	Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo		
2	Using Packet Tracer, create a basic network of two computers using appropriate network wire. Use Static IP address allocation and show connectivity		
3	Using Packet Tracer, create a basic network of One server and two computers using appropriate network wire. Use Dynamic IP address allocation and show connectivity		
4	Using Packet Tracer, create a basic network of One server and two computers and two mobile / movable devices using appropriate network wire. Show connectivity		
5	Using Packet Tracer, create a network with three routers with RIPv1 and each router associated network will have minimum three PC. Show Connectivity		
6	Using Packet Tracer, create a network with three routers with RIPv2 and each router associated network will have minimum three PC. Show Connectivity		
7	Using Packet Tracer, create a network with three routers with OSPF and each router associated network will have minimum three PC. Show Connectivity		
8	Using Packet Tracer, create a network with three routers with BGP and each router associated network will have minimum three PC. Show Connectivity		
9	Using Packet Tracer, create a wireless network of multiple PCs using appropriate access point.		
10	Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working		

Course Code	Course Title	Credits	Lectures /Week
USCS403	Software Engineering	2	3
<p><b>About the Course:</b>  This course covers a collection of methods which embody an "engineering" approach to the development of software. It discusses the nature of software and software projects, software development models, software process maturity, project planning, management, and estimations. It also underlines the topics on software testing and quality assurance.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To learn and understand the Concepts of Software Engineering</li> <li>• To learn and understand Software Development Life Cycle</li> <li>• To apply the project management and analysis principles to software project development.</li> <li>• To apply the design &amp; testing principles to software project development.</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Plan a software engineering process life cycle, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements</li> <li>• Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology.</li> <li>• Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice</li> <li>• Able to use modern engineering tools necessary for software project management, time management and software reuse.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> The Nature of Software, Software Engineering, Professional Software Development, Layered Technology, Process framework, CMM, Process Patterns and Assessment Prescriptive Models: Waterfall Model, Incremental, RAD Models Evolutionary Process Models: Prototyping, Spiral and Concurrent Development Model Specialized Models: Component based, Aspect Oriented development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming</p> <p><b>Requirement Analysis and System Modeling:</b> Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS, Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram</p>	15	

<p style="text-align: center;"><b>II</b></p>	<p><b>System Design:</b> System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design</p> <p><b>Software Measurement and Metrics:</b> Process Metrics and Project Metrics, Software Measurement, Object Oriented Metrics, Software Project Estimation, Decomposition Techniques, LOC based, FP based and Use case based estimations, Empirical estimation Models</p> <p><b>Software Project Management: Estimation</b> in Project Planning Process</p> <p>–Software Scope and Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision</p> <p><b>Project Scheduling</b> - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts</p>	<p style="text-align: center;"><b>15</b></p>
<p style="text-align: center;"><b>III</b></p>	<p><b>Risk Management</b> - Risk strategies, Software risks, Risk Identification, projection, RMMM Quality Concepts</p> <p><b>Software Quality Assurance</b> SQA activities, Software reviews, FTR, Software reliability and measures, SQA plan Software Configuration Management, elements of SCM, SCM Process, Change Control Capability Maturity Model</p> <p><b>Software Testing :</b> Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Software Engineering, A Practitioner’s Approach, Roger S, Pressman, 2019</li> <li>2. Software Engineering: principles and Practices, Deepak Jain, OXFORD University Press, 2008</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Software Engineering, Ian Sommerville, Pearson Education, 2017</li> <li>2. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI, 2018</li> <li>3. Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley &amp; Sons, 2010</li> <li>4. A Concise Introduction to Software Engineering, Pankaj Jalote, Springer</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP403	Software Engineering – Practical	1	3
Perform the following exercises for any two projects given in the list of sample projects or any other projects:			
1	Write down the problem statement for a suggested system of relevance		
2	Perform requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.		
3	Draw the function oriented diagram: Data Flow Diagram (DFD) and Structured chart.		
4	Draw the user's view analysis for the suggested system: Use case diagram.		
5	Draw the structural view diagram for the system: Class diagram, object diagram.		
6	Draw the behavioral view diagram : State-chart diagram, Activity diagram		
7	Draw the behavioral view diagram for the suggested system: Sequence diagram, Collaboration diagram		
8	Draw the implementation and environmental view diagram: Component diagram, Deployment diagram		
9	Perform Estimation of effort using FP Estimation		
10	Prepare time line chart/Gantt Chart/PERT Chart		
11	Develop test cases for unit testing and integration testing		
12	Develop test cases for various white box and black box testing		
<b>List of sample projects</b> <ol style="list-style-type: none"> <li>a. Student Result Management System</li> <li>b. Library management system</li> <li>c. Inventory control system</li> <li>d. Accounting system</li> <li>e. Fast food billing system</li> <li>f. Bank loan system</li> <li>g. Blood bank system</li> <li>h. Railway reservation system</li> <li>i. Automatic teller machine</li> <li>j. Video library management system</li> <li>k. Hotel management system</li> <li>l. Hostel management system</li> <li>m. Share online trading</li> <li>n. Hostel management system</li> <li>o. Resource management system</li> <li>p. Court case management system</li> </ol>			

Course Code	Course Title	Credits	Lectures /Week
USCS404	IoT Technologies	2	3
<p><b>About the Course:</b> The course aims to provide basic understanding of SoC architectures; IoT, different types of IoT platforms and different types of applications that can be built.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>● Introduce concepts of SoC and IoT</li> <li>● Introduce various types of IoT platforms</li> <li>● Interfacing various types of devices using different protocols with IoT</li> <li>● Understand practical applications of IoT in real life world</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>● understand SoC and IoT</li> <li>● use different types of IoT Platforms and interfaces</li> <li>● understand and implement an idea of various types of applications built using IoT</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Fundamentals of IoT:</b> Introduction, Definitions &amp; Characteristics of IoT, IoT Architectures, Physical &amp; Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.</p> <p><b>System on Chip:</b> What is System on chip? Structure of System on Chip. SoC Elements: FPGA, GPU, APU, Compute Units.</p> <p><b>Different types of IoT/SoC Platforms:</b> Introduction to Raspberry Pi, Arduino &amp; NodeMCU, Introduction to SoC-ARM Architecture, atmega328 architecture</p>	15	
II	<p><b>Interfacing with IoT Platforms:</b> Basic hardware components like LED, Button, Camera, 8X8 LED Grid, Motor etc and interfacing them for input/output with IoT devices using PWM, UART, GPIO, I2C, SPI</p> <p><b>Using Sensor &amp; Actuators:</b> Overview of Sensors working, Analog and Digital Sensors, Interfacing of Temperature, Humidity, Motion, Light and Gas Sensor, Level Sensors, Ultrasonic sensors, Interfacing of Actuators, Interfacing of Relay Switch and Servo Motor</p> <p><b>IoT and Protocols IoT Security:</b> HTTP, UPnP, CoAP, MQTT, XMPP, Privacy and Security Issues in IoT.</p>	15	



<b>III</b>	<p><b>IoT &amp; Web:</b> Web server for IoT, Sending/Receiving data between web server &amp; IoT device, Cloud for IoT, Node RED, M2M vs IoT Communication Protocols, Basics of WSNs, WSN architecture and types,</p> <p><b>IoT Applications:</b> Modern IoT case studies / applications used in the areas of transportation, agriculture, health care etc</p> <p><b>Edge Computing:</b> Edge computing purpose and definition, Edge computing use cases, Edge computing hardware architectures, Edge platforms, Edge vs Fog Computing, Communication Models - Edge, Fog and M2M.</p>	<b>15</b>
------------	---	-----------

**Textbooks:**

1. Introduction to IoT Paperback by Sudip Misra , Anandarup Mukherjee , Arijit Roy , Cambridge Press, 2022
2. Jain, Prof. Satish, Singh, Shashi, “Internet of Things and its Applications”, 1st Edition, BPB, 2020.
3. Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, Internet of Things, Wiley, India, 2019
4. IoT and Edge Computing for Architects - Second Edition, by Perry Lea, Publisher: Packt Publishing, 2020

**Additional References:**

1. Internet of Things by Vinayak Shinde, SYBGEN Learning India Pvt. Ltd, 2020
2. Internet of things, Dr. Kamlesh Lakhwani, Dr. Hemant kumar Gianey, Josef Kofi Wireko, Kamalkant Hiran, BPB Publication, 2020
3. Arduino, Raspberry Pi, NodeMCU Simple projects in easy way by Anbazhagan k and Ambika Parameswari k, 2019.
4. IoT based Projects: Realization with Raspberry Pi, NodeMCU Paperback – February 2020, by Rajesh Singh Anita Gehlot, 2020
5. Mastering the Raspberry Pi, Warren Gay, Apress, 2014

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP404</b>	<b>IoT Technologies – Practical</b>	<b>1</b>	<b>3</b>
1	Preparing Raspberry Pi: Hardware preparation and Installation		
2	Demonstrate Arduino Uno and its pins interfacing with IDE.		
3	GPIO: Light the LED with Python with/without a button using either Uno/Raspberry Pi.		
4	SPI: Camera Connection and capturing Images/Videos using SPI		
5	GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas		
6	Stepper Motor Control: PWM to manage stepper motor speed using Uno/Raspberry Pi.		
7	Node RED: Connect LED to Internet of Things		
8	Use different types of sensors (LDR, Temperature) with Raspberry Pi/Uno.		
9	Trigger a set of led GPIO on any IoT platform via any related web server		
10	Interface with any sensor and send its value over the internet to the server using any suitable protocol		

Course Code	Course Title	Credits	Lectures /Week
USCS405	Android Application Development	2	3
<p><b>About the Course:</b>  This course is aimed at creating a skilled IT workforce that is focused on developing Apps for mobile and smart Android-based computing platforms. It familiarises the development of android applications using Kotlin for problems that address real-life needs ranging from intuitive UI to rich multimedia experience.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Kotlin Programming Language for application development</li> <li>• Creating robust mobile applications on simulators and physical devices</li> <li>• Creating intuitive, reliable mobile apps using the android services and components</li> <li>• Handling data local and remote data storage</li> <li>• Create a seamless user interface that works with different mobile screens</li> </ul>			
<p><b>Learning Outcomes:</b>  After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Build useful mobile applications using Kotlin language on Android</li> <li>• Install and configure Android Studio for application development</li> <li>• Master basic to intermediate concepts of Kotlin required for mobile application development</li> <li>• Use built-in widgets and components, work with the database to store data</li> <li>• Master key Android programming concepts and deploy the application on Google Play</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Kotlin:</b> Basics of Kotlin, type conversions, comments, Kotlin operators, variables in Kotlin, packages, visibility modifiers, control flow statements, Concept of OOPS in Kotlin, classes in Kotlin, delegation and extension functions, the companion object,</p> <p><b>Advanced Concepts in Kotlin:</b> declaring and calling functions, parameters, and arguments in Kotlin, default argument, variable number of arguments, unit-returning function, explicit return type, lambda expression, coroutines, Collections in Kotlin, Mutable and Immutable Collections, Ranges, type Checks, casting concept, this expression, Null safety, exception handling, annotations</p> <p><b>App Development with Android Studio:</b> Android Architecture, Android Application Framework, Android Virtual Device, Creating and running First Android Application, working with Physical Android Device, Adding Kotlin Files in Android Studio</p> <p><b>Basics Of Android- Application Components:</b> Activities, Intent, and Broadcast Receiver, Services, Fragment, Activity Life Cycle, Content Provider, Widgets, and Notifications</p>	15	

<p style="text-align: center;"><b>II</b></p>	<p><b>Designing Android UI:</b> User Interface (UI), Layout and Its Types, Layout Attribute, working with Views, Android UI Controls, Styles and Themes, Event Handler, setting up themes in Manifest and from the application, dialog in activity, using intents, fragments</p> <p><b>Handle Images, Listview And Menu:</b> ImageView, ImageSwitcher, ListView, Menu, and its types, Designing menu in XML, Option menu, Context menu, popup menu, <b>Screen</b> Navigation, RecyclerView, Interaction of Views</p> <p><b>Data binding in Android-</b>AdapterView, Spinner, Gallery view, AutotextCompleteView, screen orientation, Design the view dynamically</p> <p><b>Implementing Data Persistence:</b> Data Storage-Shared Preference, Internal And External Storage Storing Data Using SQLite Databases, Content Provider, Firebase Real-Time Data</p>	<p style="text-align: center;"><b>15</b></p>
<p style="text-align: center;"><b>III</b></p>	<p><b>Graphics, Animations, and Integrating Media in Android:</b> Drawable Class, Animation in Android, MediaPlayer API and in Android, MediaPlayer and AudioManager Class,</p> <p><b>Interacting With Camera</b> and input gestures: Android Camera, Input gestures-multiple touch, swipe, drag, scroll, zoom, Recording</p> <p><b>Gathering Location Data:</b></p> <p><b>Managing Background Tasks:</b> Broadcast Receivers, Services, Threads and Process, AsyncTask, JobScheduler, Manage device Awake State</p> <p><b>Deploying Android applications on Google Play-</b>Publishing/Deploy the application, Versioning, signing Application</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>How to Build Android Apps with Kotlin: A hands-on guide to developing, testing, and publishing your first apps with Android, Alex Forrester, Packt Publishing, 2021</li> <li>Android Programming: Crafting UI/UX using Kotlin, SYBGEN Learning, 2020</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin Dawn Griffiths, 3rd Edition, O'Reilly Media, 2021</li> <li>Android Studio 4.2 Development Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 4.2, Kotlin and Android Jetpack, Neil Smyth, Payload Media, 2021</li> <li>Android Programming with Kotlin for Beginners, John Horton, Packt Publishing, 2019</li> <li>Android Development with Kotlin: Enhance your skills for Android development using Kotlin, Marcin Moskala, Packt Publishing</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP405</b>	<b>Android Application Development – Practical</b>	<b>1</b>	<b>3</b>
1	<ul style="list-style-type: none"> <li>i. Write a program using Kotlin to implement control structures and loops.</li> <li>ii. Write a program to implement object-oriented concepts in Kotlin.</li> </ul>		
2	<ul style="list-style-type: none"> <li>i. Create an Android application to design screens using different layouts and UI including Button, Edittext, Textview, Radio Button etc.</li> <li>ii. Write an android application demonstrating response to event/user interaction for <ul style="list-style-type: none"> <li>a. Checkbox</li> <li>b. Radio button</li> <li>c. Button</li> <li>d. Spinner</li> </ul> </li> </ul>		
3	<ul style="list-style-type: none"> <li>i. Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image.</li> <li>ii. Create an application to use Gridview for shopping cart application.</li> </ul>		
4	<ul style="list-style-type: none"> <li>i. Create an Android application to demonstrate implicit and explicit intents</li> <li>ii. Create an application to demonstrate shared preferences</li> </ul>		
5	<ul style="list-style-type: none"> <li>i. Create an Android application to demonstrate the use of Broadcast listeners.</li> <li>ii. Create an Android application to create and use services.</li> </ul>		
6	<ul style="list-style-type: none"> <li>i. Create an Android application to demonstrate XML based animation</li> <li>ii. Create an Android application to display canvas and allow the user to draw on it.</li> </ul>		
7	<ul style="list-style-type: none"> <li>i. Create a media player application in android that plays audio. Implement play, pause, and loop features.</li> <li>ii. Create an Android application to use a camera and capture image/video and display them on the screen.</li> </ul>		
8	<ul style="list-style-type: none"> <li>i. Create an android application to implement Asynctask and threading concepts.</li> <li>ii. Create an Android application to demonstrate the different types of menus. <ul style="list-style-type: none"> <li>a. Pop-up Menu</li> <li>b. Context Menu</li> <li>c. Option Menu</li> </ul> </li> </ul>		
9	Create an Android application to record the current location. Based on the current location allow the user to use some useful services/applications		
10	Create a suitable Android application to store and retrieve data in the SQLite database.		
11	Create a suitable Android application to work with Firebase for storing and manipulating data.		

Course Code	Course Title	Credits	Lectures /Week
USCS406	Advanced Application Development	2	3
<p><b>About the Course:</b> The course aims at developing scalable, robust, and maintainable web applications using MEAN stack and developing advanced mobile applications using Flutter</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To understand all the necessary and important technologies such as MongoDB, Express.js, AngularJS, and Node.js.</li> <li>To understand modern app development using Flutter</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Store the data in NoSQL, document-oriented MongoDB database that brings performance and scalability.</li> <li>Use Node.js and Express Framework for building fast, scalable network applications</li> <li>Use AngularJS framework that offers declarative, two-way data binding for web applications.</li> <li>Integrate the front-end and back-end components of the MEAN stack.</li> <li>Develop robust mobile applications using Flutter.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Node.js (N):</b> Introduction to Node.js. Installing Node.js. The package.json File. The Node.js Event Loop. The I/O Cycle. The Anatomy of a Node.js Module. Creating Node Modules. Exploring the Node.js HTTP Module. Creating an HTTP Webserver with Node.js. Responding to HTTP Requests. Routing in Node.js. Creating a Sample Node.js Application.</p> <p><b>MongoDB(M):</b> Introduction to MongoDB. Installing MongoDB. Using MongoDB Compass. Using Mongo Shell Interface. Connecting to MongoDB. Creating Schemas and Models. Querying Documents Using find(). Inserting Documents Using create(). Updating Documents Using findOneAndUpdate(). Deleting Documents Using findOneAndDelete() &amp; deleteMany()</p>	15	
II	<p><b>Server-Side Development with Express (E):</b> Introduction to the Express Framework. Installing and Testing Express. Creating a Node.js Express App. Restructuring an Express App. Creating Templates. Using Express Middleware Functions. Creating the List Page. Creating the Details Page. Creating the Edit Page. Creating the Add Page. Deleting Data. REST API Basics. Testing REST APIs. Refactoring APIs.</p> <p><b>Understanding Angular.JS(A):</b> Getting Started with Angular. Creating an Angular Application. Angular Project File Structure. Anatomy of an Angular</p>	15	

	Component. One-way Data Binding. Two-way Data Binding. Using NgIf Directive. Using NgForOf Directive. Angular Modules. Creating NgModules Using Angular Router. Configuring Templates. Creating Navigations. Working with Template-driven Forms. Working with Reactive Forms. Validating Form Data. Services Dependency Injection (DI). Reading Data from Database. Inserting Data into Database. Updating Data in the Database. Delete Data from Database.	
<b>III</b>	<p><b>Understanding Flutter:</b> Importance of Flutter, Flutter Framework, Android Studio, Flutter SDK, Installing and Configuring Flutter SDK.</p> <p><b>Dart Programming:</b> main() function, Dart Variables, Dart Data Types, Dart Conditional Operators, Control Flow &amp; Loops. Dart Functions - Functions, Function Structure, creating a Function, Function Returning Expression. Object-Oriented Programming (OOP) - Creating a Class, Adding Methods to Classes, Class — Getters and Setters, Class Inheritance, Abstract Class.</p> <p><b>Flutter Widgets Fundamentals:</b> Scaffold Widget, Image Widget, Container Widget, Column and Row Widgets, Icon Widget, Layouts in Flutter, Card Widget, Hot Reload and Hot Restart, Stateful and Stateless Widgets</p> <p><b>Navigation and Routing:</b> Button Widget, App Structure and Navigation, Navigate to a New Screen and Back, Navigate with Named Routes, Send and Return Data among Screens, Animate a Widget across Screens, WebView Widget in Flutter</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications by Brad Dayley, Brendan Dayley, Caleb Dayley, Pearson, 2018.</li> <li>2. Beginning Flutter: A Hands On Guide to App Development by Marco L. Napoli, Wrox, 2019</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Full Stack Javascript Development with Mean - MongoDB, Express, AngularJS, and Node.JS by Adam Bretz, Colin J Ihrig, Shroff/SitePoint, 2015</li> <li>2. Practical Flutter by Zammetti Frank, Apress, 2019</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP406</b>	<b>Advanced Application Development – Practical</b>	<b>1</b>	<b>3</b>
1	Write a program to implement MongoDB data models		
2	Write a program to implement CRUD operations on MongoDB		
3	Write a program to perform validation of a form using AngularJS		
4	Write a program to create and implement modules and controllers in Angular JS		
5	Write a program to implement Error Handling in Angular JS		
6	Create an application for Customer / Students records using AngularJS		
7	Write a program to create a simple web application using Express, Node JS and Angular JS		
8	Create a simple HTML “Hello World” Project using AngularJS Framework and applying ng-controller, ng-model and expressions		
9	Create an app using Flutter for User Authentication		
10	Create an app using Flutter to implement an Image Gallery		
11	Create an app using Flutter to demonstrate the use of different layouts		
12	Create an app using Flutter to demonstrate navigation in an App		



Course Code	Course Title	Credits	Lectures /Week
USCS4071	Research Methodology	2	3
<p><b>About the Course:</b> The course aims to understand the basics research, how research problems are defined, research methods are adopted and/or developed, research is undertaken, and how research results are communicated to the peers.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• The research methodology course is proposed to assist students in planning and carrying out research projects.</li> <li>• The students are exposed to the principles, procedures and techniques of implementing research project.</li> <li>• The course starts with an introduction to research and carries through the various methodologies involved.</li> <li>• It continues with finding out the literature using technology, basic statistics required for research and finally report writing.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Define research, formulate problem and describe the research process and research methods.</li> <li>• Understand and apply basic research methods including research design, data analysis and interpretation.</li> <li>• Understand ethical issues in research, write research report, research paper and publish the paper.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Research Methodology:</b> Meaning of Research, Objectives of Research, Motivations in Research, types of Research, Research Approaches, Significance of Research, Research Methods v/s Methodology, Research and Scientific Methods, Research Process, Criteria of Good Research.</p> <p><b>Defining the Research Problem:</b> Concept and need, Identification of Research problem, defining and delimiting Research problem.</p> <p><b>Formulating a Research Problem:</b> Reviewing Literature, formulating a Research Problem, Research Question, Identifying Variables, Constructing Hypothesis</p> <p><b>The Research Design:</b> Meaning, Need for Research Design, Important Concepts, Different Research Designs, Basic Principles of Experimental Designs.</p>	15	
II	<p><b>Tools for Data Collection:</b> Collections of Primary Data, Collection of Data through questionnaire and Schedules, other Observation Interview Methods,</p>	15	

	<p>Collection of Secondary Data, Selection of appropriate method for data collection, Case Study, Focus Group Discussion, Techniques of developing research tools, viz. Questionnaire and rating scales etc. Reliability and validity of Research tools.</p> <p><b>Sampling Design:</b> Steps in Sampling Design, Criteria of Selecting a Sampling Procedure, Characteristics of a Good Sample Design, Different Types of Sample Designs, how to Select a Random Sample. Probability and Non-Probability sampling types and criteria for selection, Developing sampling Frames.</p> <p><b>Overview of Hypothesis Testing:</b> What is a Hypothesis? Characteristics of good Hypothesis. Basic Concepts, Procedure for Hypothesis Testing, Flow Diagram for Hypothesis Testing, Tests of Hypotheses, and One sided and two-sided hypothesis, Type – I and Type – II errors, Null Hypothesis-Alternative Hypothesis.</p>	
<p style="text-align: center;"><b>III</b></p>	<p><b>Technical Writing:</b> Writing a Research Proposal, what is a Scientific Paper? Ethics in Scientific Publishing.</p> <p><b>Preparing the Text:</b> How to Prepare the Title, how to List the Authors and Addresses, how to Prepare the Abstract, how to Write the Introduction, how to Write the Materials and Methods Section, how to Write the Results, how to Write the Discussion, how to State the Acknowledgments, how to Cite the References.</p> <p><b>Preparing the Tables and Figures:</b> How to Design Effective Tables, how to Prepare Effective Graphs, how to Prepare Effective Photographs.</p> <p><b>Publishing the Paper:</b> Rights and Permissions, How to Submit the Manuscript, How and When to Use Abbreviations, How to Write a thesis, Outcome of Research, Ethical issues in research</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Kothari C.R., Research Methodology, New Age International Publication, 2019</li> <li>2. Research Methodology-A Step-by-Step Guide for Beginners, (4th ed.), Ranjit Kumar, Singapore, Pearson Education, 2018</li> <li>3. Research Methodology, Vaishali Khairnar, Staredu Solutions India Pvt Ltd, 2020</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Research Methodology: Methods and Techniques, Dr. R. K. Jain, , Fifth Edition, VEI, 2021</li> <li>2. Research Methodology, R. Panneerselvam, Second Edition, PHI, 2014</li> <li>3. Dr. Rachna Jain, Research Methodology, Maximax Publishing House</li> <li>4. How to Write and Publish a Scientific Paper, Cambridge University Press, Barbara Gastel and Robert A. Day, 2017</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCS4072	Management & Entrepreneurship	2	3
<p><b>About the Course:</b> The aim of the course is to develop conceptual understanding of management and administration, and comprehend the environment of making of an entrepreneur. The course focuses on giving students the business management and innovation skills required to succeed in a startup</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>To understand the idea of management, process and its levels.</li> <li>To understand the perception of entrepreneurship, process and its types.</li> <li>To understand the concept SSI and steps to start SSI.</li> <li>To understand the selection of project, project report, project appraisal, and its feasibility.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>Understand the meaning of management, functions, administration and its process.</li> <li>Understand the foundation of entrepreneurship and its theory, types and its process.</li> <li>Identify the steps involved in an entrepreneurial venture (SSI).</li> <li>Understand an entrepreneur is converting his business ideas into running concern by selecting the project.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction: Meaning,</b> Meaning, Characteristics of Management, Nature of Management, Management Functions, Functional Areas of Management, Management and Administration, Role of Management, Levels of Management, Evolution of Management</p> <p><b>Planning:</b> Meaning, Nature, importance, types of planning, types of plans, planning process, decision-making.</p> <p><b>Organizing and staffing:</b> Meaning and Definitions of Organizing, Steps in Organizing, Nature of Organization, Organization Structure, Purpose of Organization, Principles of Organization, Departmentation, Types of Organization, Span of Control, Authority, Power and Responsibility, Delegation of Authority, Centralization and Decentralization, Delegation vs Decentralization, Management by Objectives [MBO], Meaning of Staffing, Nature and Importance of Staffing, Recruitment, Selection.</p>	15	
II	<p><b>Directing and Controlling:</b> Meaning and Nature of Direction, Principles of Directing, Leadership and Leadership Styles, Motivation, Communication, Noise and Feedback in Communication, Importance of Communication, Channels of Communication, Types of Communication, Forms of Communication, Coordination, Coordination and Cooperation, Importance</p>	15	

	<p>of Coordination, Techniques of Coordination, Managerial Control, Steps in a Control Process, Essentials of a Sound Control System, Control Methods.</p> <p><b>Entrepreneurship:</b> Evolution of Concept of Entrepreneur, Concept of Entrepreneur, Characteristics of Entrepreneur, Distinction between Entrepreneur and Manager, Technical Entrepreneur, Charms of Being an Entrepreneur, Functions of an Entrepreneur, Types of Entrepreneurs, Intrapreneurs, Ultrapreneurs, Concept of Entrepreneurship, Evolution of Entrepreneurship, Role of Entrepreneurship in Economic Development, Stages in the Entrepreneurial Process, Barriers to Entrepreneurship</p>	
<b>III</b>	<p><b>Small Scale Industry:</b> Meaning and Definition of Small-Scale Industry, Characteristics of SSI, Objectives, Scope, Role of SSI in Economic Development, Advantages of Small-Scale Industries, steps to Start an SSI, Government Policy towards SSI</p> <p><b>Preparation of Project:</b> Meaning, Project Classification, Project Identification, Project Report and its significance, Contents of a Project Report, Formulation of Project Report, Planning Commission Guidelines, Network Analysis, Common Mistakes by Entrepreneurs in Project Formulation, Project Appraisal, Identification of Opportunity, Project Feasibility study.</p>	<b>15</b>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Havinal Veerabhadrapa, Management and Entrepreneurship, New Age International Publishers.</li> <li>2. Kanishka Bedi, Management and Entrepreneurship, Oxford University Press</li> <li>3. Dr. R. K. Singal, Entrepreneurship Development and Management</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. P. N. Singh, J. C. Saboo, Entrepreneurship Management, 6th Edition, Dr. P. N. Singh Centre for Hrd Publications.</li> <li>2. Donald L. Sexton &amp; Raymond W. Smilor, The Art and Science of Entrepreneurship, Ballinger, 2022</li> <li>3. Clifford M. Baumback &amp; Joseph R. Mancuso, Entrepreneurship And Venture Management, Prentice Hall</li> </ol>		

## Evaluation Scheme

### I. Internal Evaluation for Theory Courses – 25 Marks

#### (i) Mid-Term Class Test – 15 Marks

- It should be conducted using any **learning management system** such as **Moodle** (Modular object-oriented dynamic learning environment)
- The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes**.

#### (ii) Assignment/ Case study/ Presentations – 10 Marks

- Assignment / Case Study Report / Presentation can be uploaded on any **learning management system**.

### II. External Examination for Theory Courses – 75 Marks

- Duration: **2.5 Hours**
- Theory question paper pattern:

<b>All questions are compulsory.</b>			
<b>Question</b>	<b>Based on</b>	<b>Options</b>	<b>Marks</b>
Q.1	Unit I	<i>Any 4 out of 6</i>	20
Q.2	Unit II	<i>Any 4 out of 6</i>	20
Q.3	Unit III	<i>Any 4 out of 6</i>	20
Q.4	Unit I, II and III	<i>Any 5 out of 6</i>	15

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be sub-divided into sub questions as a, b, c, d, etc. & the allocation of Marks depends on the weightage of the topic.

### III. Practical Examination

- Each core subject carries 50 Marks  
**40 marks + 05 marks (journal) + 05 marks (viva)**
- Duration: **2 Hours** for each practical course.
- Minimum **80% practical** from each core subjects are required to be completed.
- **Certified Journal is compulsory for appearing at the time of Practical Exam**
- The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

-----

**UNIVERSITY OF MUMBAI**



**Syllabus for**  
**Program: Bachelor of Science**  
**Course: Computer Science**

*with effect from*  
**Academic Year 2023-2024**

## Preamble

The revised and restructured syllabus aims to provide students with a comprehensive understanding of computer science concepts, theories, and practical skills, enabling them to excel in the dynamic and rapidly evolving field of technology. The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skill sets demanded in the new technological environment. It also endeavors to align the program structure and course curriculum with student aspirations and corporate expectations. The proposed curriculum is contextual, industry-friendly, and suitable to cater to the needs of society and the nation in the present-day environment.

The TYBSc Computer Science syllabus is structured as follows:

**Core Modules:** The syllabus comprises core modules that cover essential topics in computer science, ensuring a strong foundation that aim to develop computational thinking, analytical abilities, and problem-solving skills among students. The Artificial Intelligence course provides in-depth knowledge of cutting edge AI concepts and techniques empowering them to develop intelligent systems and algorithms. With a focus on safeguarding information and systems the Cyber and Information Security course equips students with essential concepts and practices in cybersecurity. The Data Science course provides students with a solid foundation in data analysis and interpretation, enabling them to extract valuable insights and make data-driven decisions. In an era dominated by cloud-based technologies, the Cloud Computing course focuses on the principles, architectures, and applications of cloud computing.

**Skill Enhancement Electives:** Students are required to choose skill enhancement electives to deepen their knowledge in specific areas of interest. The electives offer specialized courses such as web development, cybersecurity, data science, or software engineering. By selecting these courses, students can tailor their learning experience according to their career aspirations and industry demands. Skill Enhancement courses such as Linux Server Administration, Software Testing and Quality Assurance, Cyber Forensics, Game Programming, Data Mining and Warehousing, Wireless and Sensor Networks, Ethical Hacking, and Information Retrieval cater to specialized areas of expertise and industry demands.

**Generic Electives:** The syllabus also includes generic electives, which provide students with the option to explore disciplines of interest beyond their choices in Core and Discipline-Specific Elective papers. These courses broaden their horizons and allow for interdisciplinary learning.

**Project Work:** A significant component of the syllabus involves hands-on project work. Through practical assignments and projects, students have the opportunity to apply their theoretical knowledge to real-world scenarios. This approach fosters creativity, problem-solving skills, and innovation in designing and developing software solutions.

Assessment methods for the TY Computer Science syllabus include written examinations, practical assignments, project evaluations, and presentations. This comprehensive approach ensures that students' understanding is evaluated through various mediums, emphasizing both theoretical knowledge and practical skills.

The newly designed TY Computer Science syllabus aims to equip students with the necessary competencies to pursue careers in software development, data analysis, research, or further studies in computer science-related disciplines. It seeks to empower students with the knowledge and skills required to thrive in the ever-evolving landscape of technology and contribute to the advancement of the field.

We sincerely believe that students who undertake this program will gain a strong foundation and exposure to the basics, advanced concepts, and emerging trends in the subject. We express our gratitude to all the experts who provided valuable feedback and suggestions to improve the curriculum. We have made sincere efforts to incorporate their inputs. Special appreciation goes to the University Department of Computer Science and colleagues from various colleges who volunteered or indirectly contributed to designing certain specialized courses and the syllabus as a whole.

# T.Y.B.Sc. Computer Science Syllabus

## Choice Based Credit System (CBCS)

*with effect from*

**Academic year 2023-2024**

<b>Semester – V</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
USCS501	Core Subject	Artificial Intelligence	3	3
USCSP501	Core Subject Practical	Artificial Intelligence – Practical	1	3
USCS502	Core Subject	Information & Network Security	3	3
USCSP502	Core Subject Practical	Information & Network Security – Practical	1	3
USCS5031	Skill Enhancement Elective 1* (SEE)	Linux Server Administration	3	3
USCSP5031	Skill Enhancement Elective 1* Practical (SEEP)	Linux Server Administration – Practical	1	3
USCS5032	Skill Enhancement Elective 1* (SEE)	Software Testing & Quality Assurance	3	3
USCSP5032	Skill Enhancement Elective 1* Practical (SEEP)	Software Testing & Quality Assurance – Practical	1	3
USCS5041	Skill Enhancement Elective 2* (SEE)	Cyber Forensics	3	3
USCSP5041	Skill Enhancement Elective 2* Practical (SEEP)	Cyber Forensics – Practical	1	3
USCS5042	Skill Enhancement Elective 2* (SEE)	Game Programming	3	3
USCSP5042	Skill Enhancement Elective 2* Practical (SEEP)	Game Programming – Practical	1	3
USCS5051	Generic Elective**	Project Management	2	3
USCS5052	Generic Elective**	Operations Research	2	3
USCSP505	Project	Project Work – I	2	3

\* One course each from Skill Enhancement Elective 1 and Skill Enhancement Elective 2 should be selected by the student.

\*\* One course from Generic Elective should be selected by the student



# T.Y.B.Sc. Computer Science Syllabus

## Choice Based Credit System (CBCS)

*with effect from*

**Academic year 2023-2024**

<b>Semester – VI</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
USCS601	Core Subject	Data Science	3	3
USCSP601	Core Subject Practical	Data Science – Practical	1	3
USCS602	Core Subject	Cloud Computing and Web Services	3	3
USCSP602	Core Subject Practical	Cloud Computing and Web Services – Practical	1	3
USCS6031	Skill Enhancement Elective 1* (SEE)	Wireless and Sensor Networks	3	3
USCSP6031	Skill Enhancement Elective 1* Practical (SEEP)	Wireless and Sensor Networks – Practical	1	3
USCS6032	Skill Enhancement Elective 1* (SEE)	Information Retrieval	3	3
USCSP6032	Skill Enhancement Elective 1* Practical (SEEP)	Information Retrieval – Practical	1	3
USCS6041	Skill Enhancement Elective 2* (SEE)	Data Mining & Warehousing	3	3
USCSP6041	Skill Enhancement Elective 2* Practical (SEEP)	Data Mining & Warehousing – Practical	1	3
USCS6042	Skill Enhancement Elective 2* (SEE)	Ethical Hacking	3	3
USCSP6042	Skill Enhancement Elective 2* Practical (SEEP)	Ethical Hacking – Practical	1	3
USCS6051	Generic Elective**	Customer Relationship Management	2	3
USCS6052	Generic Elective**	Cyber Laws and IPR	2	3
USCSP605	Project	Project Work – II	2	3

\* One course each from Skill Enhancement Elective 1 and Skill Enhancement Elective 2 should be selected by the student.

\*\* One course from Generic Elective should be selected by the student

## Semester V

Course Code	Course Title	Credits	Lectures /Week
USCS501	Artificial Intelligence	2	3
<p><b>About the Course:</b> This course provides an introduction to the field of Artificial Intelligence (AI) and explores various topics related to intelligent agents, problem-solving, knowledge representation, reasoning, machine learning, and probabilistic models. The course covers both theoretical concepts and practical applications of AI techniques. Students will gain a solid foundation in AI and develop the skills to design and implement intelligent systems.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the foundations, history, and state of the art of AI.</li> <li>• Learn about intelligent agents, their environments, and the structure of agents.</li> <li>• Explore different problem-solving strategies, including uninformed and informed search techniques.</li> <li>• Gain knowledge of knowledge representation and reasoning methods, and apply them to solve complex problems.</li> <li>• Develop an understanding of machine learning techniques, including classification, regression, and ensemble learning.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Demonstrate knowledge of the foundations and key concepts in the field of AI.</li> <li>• Analyze and design intelligent agents for specific environments.</li> <li>• Apply problem-solving techniques and algorithms to find solutions to different types of problems.</li> <li>• Construct knowledge representation models and use reasoning techniques to derive new knowledge.</li> <li>• Implement machine-learning algorithms and evaluate their performance for classification and regression tasks.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to AI and Intelligent Agents</b></p> <p><b>What Is AI:</b> Foundations, History and State of the Art of AI</p> <p><b>Intelligent Agents:</b> Agents and Environments, Nature of Environments, Structure of Agents.</p> <p><b>Problem Solving by searching:</b> Problem-Solving Agents, Uninformed Search Strategies, Informed (Heuristic) Search Strategies</p>	15	
II	<b>Knowledge Representation, Reasoning, and Machine Learning</b>	15	

	<p><b>Knowledge Representation and Reasoning:</b> Knowledge Representation and different forms, Reasoning, Planning, Uncertainty in Knowledge Fuzzy Logic &amp; Fuzzification</p> <p><b>Machine Learning:</b> Forms of Learning, Parametric &amp; Non-Parametric Models, Classification, Regression, Regularization, Decision Trees, SVM, Artificial Neural Networks, Ensemble Learning, Boosting, K-NN, Gradient Descent</p>	
<b>III</b>	<p><b>Probabilistic Models, Unsupervised Learning, and Reinforcement Learning</b></p> <p><b>Probabilistic models:</b> Statistical Learning, Learning with Complete Data, Naive Bayes Classifier, Learning with Hidden Variables: The EM Algorithm</p> <p><b>Unsupervised Learning:</b> Concept of Unsupervised learning,, Association Rule Mining</p> <p><b>Reinforcement learning:</b> Concept of Reinforcement learning, Q-Learning, Hidden Markov Model</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010.</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Artificial Intelligence: Foundations of Computational Agents, David L Poole, Alan K. Mackworth, 2nd Edition, Cambridge University Press, 2017.</li> <li>2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017</li> <li>3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP501	Artificial Intelligence – Practical	1	3
1	Breadth First Search & Iterative Depth First Search <ul style="list-style-type: none"> <li>• Implement the Breadth First Search algorithm to solve a given problem.</li> <li>• Implement the Iterative Depth First Search algorithm to solve the same problem.</li> <li>• Compare the performance and efficiency of both algorithms.</li> </ul>		
2	A* Search and Recursive Best-First Search <ul style="list-style-type: none"> <li>• Implement the A* Search algorithm for solving a pathfinding problem.</li> <li>• Implement the Recursive Best-First Search algorithm for the same problem.</li> <li>• Compare the performance and effectiveness of both algorithms.</li> </ul>		
3	Decision Tree Learning <ul style="list-style-type: none"> <li>• Implement the Decision Tree Learning algorithm to build a decision tree for a given dataset.</li> <li>• Evaluate the accuracy and effectiveness of the decision tree on test data.</li> <li>• Visualize and interpret the generated decision tree.</li> </ul>		
4	Feed Forward Backpropagation Neural Network <ul style="list-style-type: none"> <li>• Implement the Feed Forward Backpropagation algorithm to train a neural network.</li> <li>• Use a given dataset to train the neural network for a specific task.</li> <li>• Evaluate the performance of the trained network on test data.</li> </ul>		
5	Support Vector Machines (SVM) <ul style="list-style-type: none"> <li>• Implement the SVM algorithm for binary classification.</li> <li>• Train an SVM model using a given dataset and optimize its parameters.</li> <li>• Evaluate the performance of the SVM model on test data and analyze the results.</li> </ul>		
6	Adaboost Ensemble Learning <ul style="list-style-type: none"> <li>• Implement the Adaboost algorithm to create an ensemble of weak classifiers.</li> <li>• Train the ensemble model on a given dataset and evaluate its performance.</li> <li>• Compare the results with individual weak classifiers.</li> </ul>		
7	Naive Bayes' Classifier <ul style="list-style-type: none"> <li>• Implement the Naive Bayes' algorithm for classification.</li> <li>• Train a Naive Bayes' model using a given dataset and calculate class probabilities.</li> <li>• Evaluate the accuracy of the model on test data and analyze the results.</li> </ul>		
8	K-Nearest Neighbors (K-NN) <ul style="list-style-type: none"> <li>• Implement the K-NN algorithm for classification or regression.</li> <li>• Apply the K-NN algorithm to a given dataset and predict the class or value for test data.</li> <li>• Evaluate the accuracy or error of the predictions and analyze the results.</li> </ul>		

9	<p>Association Rule Mining</p> <ul style="list-style-type: none"><li>• Implement the Association Rule Mining algorithm (e.g., Apriori) to find frequent itemsets.</li><li>• Generate association rules from the frequent itemsets and calculate their support and confidence.</li><li>• Interpret and analyze the discovered association rules.</li></ul>
10	<p>Demo of OpenAI/TensorFlow Tools</p> <ul style="list-style-type: none"><li>• Explore and experiment with OpenAI or TensorFlow tools and libraries.</li><li>• Perform a demonstration or mini-project showcasing the capabilities of the tools.</li><li>• Discuss and present the findings and potential applications.</li></ul>

Course Code	Course Title	Credits	Lectures /Week
USCS502	Information & Network Security	2	3
<p><b>About the Course:</b> This course provides an in-depth understanding of the principles and techniques used in computer and network security. Students will explore various security topics, including encryption techniques, public-key cryptography, key management, message authentication, digital signatures, authentication protocols, network security, web security, intrusion detection, malicious software, and firewall design principles. Through theoretical learning and practical exercises, students will develop the necessary knowledge and skills to analyze, design, and implement secure systems and protect against security threats.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Familiarize students with the fundamental principles, models, and mechanisms of computer and network security.</li> <li>• Explore various encryption techniques, including symmetric and public-key cryptography, and understand their strengths, weaknesses, and real-world applications.</li> <li>• Examine different authentication and key management methods to ensure secure communication and protect against unauthorized access.</li> <li>• Understand the concepts and techniques of message authentication, digital signatures, and authentication protocols used in secure communication systems.</li> <li>• Investigate network security measures, including IP security, web security, intrusion detection, malicious software detection, and firewall design principles.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to:</p> <ul style="list-style-type: none"> <li>• Analyze and evaluate security trends, attacks, and mechanisms, and propose effective security solutions based on the OSI security architecture.</li> <li>• Apply classical encryption techniques, such as substitution and transposition ciphers, to encrypt and decrypt messages and analyze their security implications.</li> <li>• Implement public-key cryptography algorithms, including RSA, and demonstrate the ability to securely exchange keys and establish secure communication channels.</li> <li>• Design and implement secure authentication mechanisms, including message authentication codes and digital signatures, to ensure data integrity and non-repudiation.</li> <li>• Evaluate and implement various security measures, such as IP security, web security protocols (e.g., SSL/TLS), intrusion detection systems, and firewall configurations, to protect networks and systems from unauthorized access and attacks.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms</p> <p><b>Classical Encryption Techniques:</b> Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round</p>	15	

	<p>details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers</p> <p><b>Public-Key Cryptography and RSA:</b> Principles of Public-Key Cryptosystems, The RSA Algorithm</p>	
<b>II</b>	<p><b>Key Management:</b> Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange</p> <p><b>Message Authentication and Hash Functions:</b> Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC</p> <p><b>Digital Signatures and Authentication:</b> Digital Signatures, Authentication Protocols, Digital Signature Standard</p> <p><b>Authentication Applications:</b> Kerberos, X.509 Authentication, Public-Key Infrastructure</p>	<b>15</b>
<b>III</b>	<p><b>Electronic Mail Security:</b> Pretty Good Privacy, S/MIME</p> <p><b>IP Security:</b> Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management</p> <p><b>Web Security:</b> Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction</p> <p><b>Intrusion:</b> Intruders, Intrusion Techniques, Intrusion Detection</p> <p><b>Malicious Software:</b> Viruses and Related Threats, Virus Countermeasures, DDOS</p> <p><b>Firewalls:</b> Firewall Design Principles, Types of Firewalls</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Cryptography and Network Security: Principles and Practice 7th edition, William Stallings, Pearson</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Cryptography and Network, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH.</li> <li>2. Atul Kahate, "Cryptography and Network Security", Tata McGraw-Hill.</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP502</b>	<b>Information &amp; Network Security – Practical</b>	<b>1</b>	<b>3</b>
1	Implementing Substitution and Transposition Ciphers: Design and implement algorithms to encrypt and decrypt messages using classical substitution and transposition techniques.		
2	RSA Encryption and Decryption: Implement the RSA algorithm for public-key encryption and decryption, and explore its properties and security considerations.		
3	Message Authentication Codes: Implement algorithms to generate and verify message authentication codes (MACs) for ensuring data integrity and authenticity.		
4	Digital Signatures: Implement digital signature algorithms such as RSA-based signatures, and verify the integrity and authenticity of digitally signed messages.		
5	Key Exchange using Diffie-Hellman: Implement the Diffie-Hellman key exchange algorithm to securely exchange keys between two entities over an insecure network.		
6	IP Security (IPsec) Configuration: Configure IPsec on network devices to provide secure communication and protect against unauthorized access and attacks.		
7	Web Security with SSL/TLS: Configure and implement secure web communication using SSL/TLS protocols, including certificate management and secure session establishment.		
8	Intrusion Detection System: Set up and configure an intrusion detection system (IDS) to monitor network traffic and detect potential security breaches or malicious activities.		
9	Malware Analysis and Detection: Analyze and identify malware samples using antivirus tools, analyze their behavior, and develop countermeasures to mitigate their impact.		
10	Firewall Configuration and Rule-based Filtering: Configure and test firewall rules to control network traffic, filter packets based on specified criteria, and protect network resources from unauthorized access.		



Course Code	Course Title	Credits	Lectures /Week
USCS5031	Linux Server Administration	2	3
<p><b>About the Course:</b> This course provides an in-depth understanding of Linux server administration. Students will learn the technical aspects of various Linux distributions, software management, user and group administration, file systems, core system services, networking, security, and advanced internet services. Practical hands-on exercises and real-world examples will be used to enhance the learning experience.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Develop a solid understanding of Linux server administration principles and concepts.</li> <li>• Acquire practical skills in managing users, groups, and file systems in a Linux environment.</li> <li>• Learn how to configure and secure network services such as DNS, FTP, Apache web server, SMTP, POP, IMAP, and SSH.</li> <li>• Gain knowledge of advanced network administration topics including NFS, Samba, DFS, NIS, LDAP, DHCP, MySQL, LAMP applications, file services, email services, chat applications, and VPN.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Demonstrate proficiency in managing software packages and repositories in Linux.</li> <li>• Configure and administer user accounts, groups, and permissions in a Linux system.</li> <li>• Implement network services such as DNS, FTP, and web servers, ensuring proper security measures.</li> <li>• Design and manage advanced network services including NFS, Samba, and LDAP for efficient file sharing and user authentication.</li> <li>• Apply troubleshooting techniques to identify and resolve common issues in Linux server administration.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> Technical Summary of Linux Distributions, Managing Software</p> <p><b>Single-Host Administration:</b> Managing Users and Groups, Booting and shutting down processes. File Systems, Core System Services, Process of configuring, compiling, Linux Kernel</p> <p><b>Networking and Security:</b> TCP/IP for System Administrators, basic network Configuration, Linux Firewall (Netfilter), System and network security</p>	15	
II	<p><b>Internet Services:</b> Domain Name System (DNS), File Transfer Protocol (FTP), Apache web server, Simple Mail Transfer Protocol (SMTP), Post Office Protocol and Internet Mail Access Protocol (POP and IMAP), Secure Shell (SSH), Network authentication system (Kerberos), Domain Name Service (DNS), Security</p>	15	

<b>III</b>	<p><b>Internet Services:</b>  Network File System (NFS), Samba, Distributed File Systems (DFS), Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP), Dynamic Host configuration Protocol (DHCP), MySQL, LAMP Applications, File Services, Email Services, Chat applications, Virtual Private Networking.</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Linux Administration: A Beginner's Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016</li> <li>2. Ubuntu Server Guide, Ubuntu Documentation Team, 2016</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP5031</b>	<b>Linux Server Administration – Practical</b>	<b>1</b>	<b>3</b>
1	Install DHCP Server in Ubuntu 16.04		
2	Initial settings: Add a User, Network Settings, Change to static IP address, Disable IPv6 if not needed, Configure Service, display the list of services which are running. Stop and turn OFF auto-start setting for a service if you don't need it, Sudo Settings		
3	Configure NP Server (NTPd), Install and configure NTPd, Configure NTP Client (Ubuntu and Windows)		
4	SSH Server : Password Authentication Configure SSH server to manage a server from the remote computer, SSH Client : (Ubuntu and Windows)		
5	Install DNS server BIND, Configure DNS server which resolves domain name or IP address, Install BIND 9, Configure BIND, Limit ranges You allow to access if needed.		
6	Configure DHCP Server, Configure DHCP (Dynamic Host Configuration Protocol) Server, Configure NFS server to share directories on your NFS, Configure NFS Client. (Ubuntu and Windows Client OS)		
7	Configure LDDAP Server, Configure LDAP Server in order to share users' accounts in your local networks, Add LDAP User Accounts In the networks. Install phpLDAPadmin to operate LDAP server via Web browser.		
8	Configure NIS Server in order to share users; accounts in your local networks, Configure NIS Client to bind NIS Server.		
9	Install MySQL to configure database server, Install phpMyAdmin to operate MySQL on web browser from Clients.		
10	Install Samba to share folders or files between Windows and Linux		

Course Code	Course Title	Credits	Lectures /Week
USCS5032	Software Testing & Quality Assurance	2	3
<p><b>About the Course:</b> This course provides an in-depth understanding of software testing principles, techniques, and quality assurance practices. Students will learn the concepts and methodologies required to ensure the quality and reliability of software systems. The course covers various aspects of software testing, including test case design, verification and validation, defect management, software quality assurance, and quality improvement techniques. Students will gain hands-on experience with industry-standard tools and techniques used in software testing and quality assurance.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the significance of software testing in ensuring software quality and reliability.</li> <li>• Familiarize with the fundamental concepts and principles of software quality assurance.</li> <li>• Learn different software testing techniques and methodologies for effective test case design.</li> <li>• Explore the role of verification and validation in software development and testing processes.</li> <li>• Gain practical experience in using software testing tools and frameworks for automated testing.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Explain the importance of software testing and its impact on software quality.</li> <li>• Apply appropriate software testing techniques to identify and mitigate software defects.</li> <li>• Design and execute test cases to verify the functionality and performance of software systems.</li> <li>• Understand the principles of verification and validation and their application in software testing.</li> <li>• Utilize software testing tools and frameworks to automate testing processes and improve efficiency.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Software Testing and Quality Assurance</b></p> <p><b>Introduction to Software Testing:</b> Nature of errors and the need for testing</p> <p><b>Definition of Quality and Quality Assurance:</b> Understanding quality in software development, Distinction between Quality Assurance (QA), Quality Control (QC), Quality Management (QM), and Software Quality Assurance (SQA)</p> <p><b>Software Development Life Cycle (SDLC):</b> Overview of SDLC phases and their relationship to testing, Role of testing in each phase, Software quality factors and their impact on testing</p> <p><b>Verification and Validation (V&amp;V):</b> Definition of V&amp;V and its significance in software development, Different types of V&amp;V mechanisms, Concepts of Software Reviews, Inspection, and Walkthrough</p>	15	

<b>II</b>	<p><b>Software Testing Techniques and Strategies</b></p> <p><b>Testing Fundamentals:</b> Basics of software testing process, Test case design principles and techniques, Test execution, reporting, and documentation</p> <p><b>White Box Testing and Black Box Testing:</b> Functional/Specification based Testing as Black Box, Black box: Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition Testing. Structural Testing as White Box, White Box: Statement testing, Branch testing. Experience-based: Error guessing, Exploratory testing, Checklist-based testing.</p> <p><b>Software Testing Strategies:</b> Strategic approach to software testing Unit Testing: purpose, techniques, and best practices, Integration Testing: approaches and challenges, Validation Testing: ensuring adherence to user requirements, System Testing: comprehensive end-to-end testing</p> <p><b>Software Metrics:</b> Concept of software metrics and their importance, Developing and utilizing different types of metrics, Complexity metrics and their significance in testing</p>	<b>15</b>
<b>III</b>	<p><b>Defect Management and Software Quality Assurance</b></p> <p><b>Defect Management:</b> Definition of defects and their lifecycle, Defect management process, including defect reporting and tracking, Metrics related to defects and their utilization for process improvement</p> <p><b>Software Quality Assurance:</b> Understanding quality concepts and the Quality Movement: Background issues and challenges in SQA, Activities and approaches in Software Quality Assurance, Software Reviews: Formal Technical Reviews and their benefits, Statistical Quality Assurance and Software Reliability</p> <p><b>Statistical process control techniques for quality assurance:</b> Software reliability measurement and improvement, The ISO 9000 Quality Standards and their application in software development</p> <p><b>Quality Improvement Techniques:</b> Introduction to quality improvement methodologies, Utilizing quality costs for decision-making, Introduction to quality improvement tools: Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005</li> <li>2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall.</li> <li>2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley &amp; Sons, Inc. , Publication.</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP5032</b>	<b>Software Testing &amp; Quality Assurance – Practical</b>	<b>1</b>	<b>3</b>
1	Install Selenium IDE and create a test suite containing a minimum of 4 test cases for different web page formats (e.g., HTML, XML, JSON, etc.).		
2	Conduct a test suite for two different websites using Selenium IDE. Perform various actions like clicking links, filling forms, and verifying content.		
3	Install Selenium Server (Selenium RC) and demonstrate its usage by executing a script in Java or PHP to automate browser actions.		
4	Write a program using Selenium WebDriver to automate the login process on a specific web page. Verify successful login with appropriate assertions.		
5	Write a program using Selenium WebDriver to update 10 student records in an Excel file. Perform data manipulation and verification.		
6	Write a program using Selenium WebDriver to select the number of students who have scored more than 60 in any one subject (or all subjects). Perform data extraction and analysis.		
7	Write a program using Selenium WebDriver to provide the total number of objects present or available on a web page. Perform object identification and counting.		
8	Write a program using Selenium WebDriver to get the number of items in a list or combo box on a web page. Perform element identification and counting.		
9	Write a program using Selenium WebDriver to count the number of checkboxes on a web page, including checked and unchecked counts. Perform checkbox identification and counting.		
10	Perform load testing on a web application using JMeter. Generate and analyze load scenarios. Additionally, explore bug tracking using Bugzilla as a tool for logging and tracking software defects.		

Course Code	Course Title	Credits	Lectures /Week
USCS5041	Cyber Forensics	2	3
<p><b>About the Course:</b> This course introduces computer forensics, focusing on the techniques and methodologies used in investigating and analyzing digital evidence. Students will learn the fundamentals of computer investigations, data acquisition, crime scene processing, and the use of specialized tools for analysis. The course covers various aspects of computer forensics, including network forensics, mobile device forensics, e-mail investigations, and report writing.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the principles and concepts of computer forensics.</li> <li>• Develop skills in conducting computer investigations using a systematic approach.</li> <li>• Gain proficiency in acquiring and preserving digital evidence from different storage formats.</li> <li>• Explore the use of specialized tools and software for computer forensics analysis.</li> <li>• Learn the techniques for investigating network-related incidents and conducting live acquisitions.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Demonstrate a solid understanding of the principles and techniques used in computer forensics investigations.</li> <li>• Apply systematic approaches to acquire, preserve, and analyze digital evidence from various sources.</li> <li>• Utilize specialized tools and software for conducting effective computer forensics analysis.</li> <li>• Develop strong skills in investigating network-related incidents, including live acquisitions and network forensics.</li> <li>• Generate comprehensive and well-written reports that accurately document the findings of computer forensic investigations.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> Understanding Computer Forensics, Preparing for Computer Investigations, Maintaining Professional Conduct</p> <p><b>Computer Investigations:</b> Preparing a Computer Investigation, Taking a Systematic Approach, Procedures for Corporate High-Tech Investigations, Understanding Data Recovery Workstations and Software, Conducting an Investigation</p> <p><b>Data Acquisition:</b> Storage Formats for Digital Evidence, Determining the Best Acquisition Method, Contingency Planning for Image Acquisitions</p>	15	
II	<p><b>Processing Crime and Incident Scenes:</b> Identifying Digital Evidence, Preparing for a Search, Securing a Computer Incident or Crime Scene, Seizing Digital Evidence at the Scene, Storing Digital Evidence</p> <p><b>Current Computer Forensics Tools:</b> Evaluating Computer Forensics Tool Needs, Computer Forensics Software Tools, Computer Forensics Hardware Tools</p>	15	

	<p><b>Computer Forensics Analysis and Validation:</b> Determining What Data to Collect and Analyze, Validating Forensic Data, Addressing Data-Hiding Techniques, Performing Remote Acquisitions</p> <p><b>Recovering Graphics Files:</b> Recognizing a Graphics File, Locating and Recovering Graphics Files, Identifying Unknown File Formats</p>	
<b>III</b>	<p><b>Network Forensics and Live Acquisitions:</b> Network Forensics Overview, Performing Live Acquisitions, Developing Standard Procedures for Network Forensics, Using Network Tools</p> <p><b>E-mail Investigations:</b> Role of E-mail in Investigations, Investigating E-mail Crimes and Violations, Using Specialized E-mail Forensics Tools</p> <p><b>Cell Phone and Mobile Device Forensics:</b> Overview, Acquisition Procedures for Cell Phones and Mobile Devices</p> <p><b>Report Writing for Investigations:</b> Importance of Reports, Guidelines for Writing Reports, Generating Report Findings with Forensics Software Tools</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Bill Nelson, Amelia Philips and Christopher Steuart, "Guide to computer forensics and investigations", course technology, 6th edition</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Kevin Mandia, Chris Prosis, "Incident Response and computer forensics", Tata McGrawHill</li> </ol>		



Course Code	Course Title	Credits	Lectures /Week
USCSP5041	Cyber Forensics – Practical	1	3
1	Creating a Forensic Image using FTK Imager/Encase Imager : <ul style="list-style-type: none"> <li>• Creating Forensic Image</li> <li>• Check Integrity of Data</li> <li>• Analyze Forensic Image</li> </ul>		
2	Data Acquisition: <ul style="list-style-type: none"> <li>• Perform data acquisition using:</li> <li>• USB Write Blocker + Encase Imager</li> <li>• SATA Write Blocker + Encase Imager</li> <li>• Falcon Imaging Device</li> </ul>		
3	Analyze the memory dump of a running computer system. <ul style="list-style-type: none"> <li>• Extract volatile data, such as open processes, network connections, and registry information.</li> </ul>		
4	Capturing and analyzing network packets using Wireshark (Fundamentals) : <ul style="list-style-type: none"> <li>• Identification the live network</li> <li>• Capture Packets</li> <li>• Analyze the captured packets</li> </ul>		
5	Using Sysinternals tools for Network Tracking and Process Monitoring : <ul style="list-style-type: none"> <li>• Check Sysinternals tools</li> <li>• Monitor Live Processes</li> <li>• Capture RAM</li> <li>• Capture TCP/UDP packets</li> <li>• Monitor Hard Disk</li> <li>• Monitor Virtual Memory</li> <li>• Monitor Cache Memory</li> </ul>		
6	Recovering and Inspecting deleted files <ul style="list-style-type: none"> <li>• Check for Deleted Files</li> <li>• Recover the Deleted Files</li> <li>• Analyzing and Inspecting the recovered files</li> <li>• Perform this using recovery option in ENCASE and also Perform manually through command line</li> </ul>		
7	Steganography Detection <ul style="list-style-type: none"> <li>• Detect hidden information or files within digital images using steganography analysis tools.</li> <li>• Extract and examine the hidden content.</li> </ul>		
8	Mobile Device Forensics <ul style="list-style-type: none"> <li>• Perform a forensic analysis of a mobile device, such as a smartphone or tablet.</li> <li>• Retrieve call logs, text messages, and other relevant data for investigative purposes.</li> </ul>		

9	<p>Email Forensics</p> <ul style="list-style-type: none"><li>• Analyze email headers and content to trace the origin of suspicious emails.</li><li>• Identify potential email forgeries or tampering.</li></ul>
10	<p>Web Browser Forensics</p> <ul style="list-style-type: none"><li>• Analyze browser artifacts, including history files, bookmarks, and download records.</li><li>• Analyze cache and cookies data to reconstruct user-browsing history and identify visited websites or online activities.</li><li>• Extract the relevant log or timestamp file, analyze its contents and interpret the timestamp data to determine the user's last internet activity and associated details.</li></ul>

Course Code	Course Title	Credits	Lectures /Week
USCS5042	Game Programming	2	3
<p><b>About the Course:</b> This course provides an in-depth exploration of 3D game development and graphics, covering topics such as vectors, transformations, 3D modeling, rendering, physics-based simulation, and game engine architecture. Students will gain hands-on experience using industry-standard tools and technologies, including DirectX, Unity, and Python-Pygame. Through a combination of theoretical study, practical exercises, and project-based learning, students will develop the skills necessary to create immersive 3D games and interactive virtual worlds.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the fundamentals of vectors, transformations, and 3D graphics.</li> <li>• Develop proficiency in using industry-standard tools and technologies for 3D game development.</li> <li>• Implement advanced graphics techniques, including lighting, shading, and texturing, to create visually appealing game environments.</li> <li>• Apply principles of game design and create engaging and immersive gaming experiences.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Apply vector manipulation techniques and transformations to create and manipulate objects in 3D space.</li> <li>• Utilize industry-standard tools and technologies such as Unity and DirectX for 3D game development.</li> <li>• Implement advanced graphics techniques, including lighting, shading, and texturing, to create visually stunning game environments.</li> <li>• Design and develop games that incorporate principles of game design to create engaging and immersive experiences.</li> <li>• Deploy and showcase 3D games on various platforms, demonstrating proficiency in game development.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Vectors:</b> Vectors: Vector Manipulation, multiplying a Vector by a Scalar, Vector Addition and Subtraction, Position Vectors, Unit Vectors, Cartesian Vectors, Vector Multiplication, Scalar Product, Example of the Dot Product, The Dot Product in Lighting Calculations, The Dot Product in Back-Face Detection, The Vector Product, The Right-Hand Rule, deriving a Unit Normal Vector for a Triangle Areas, Calculating 2D Areas</p> <p><b>Transformations:</b> 2D Transformations, Matrices, Homogeneous Coordinates, 3D Transformations, Change of Axes, Direction Cosines, rotating a Point about an Arbitrary Axis, Transforming Vectors, Determinants, Perspective Projection, Interpolation</p> <p><b>3D Graphics for Game Programming:</b> 3D Transformations, Quaternions,</p>	15	

	3D Modeling and Rendering, Ray Tracing, Shader Models, Lighting, Color, Texturing, Camera and Projections, Culling and Clipping, Character Animation, Physics-based Simulation, Scene Graphs.	
<b>II</b>	<p><b>Game Engines and Design:</b> Game engine architecture, Engine support systems, Resources and File systems, Game loop and real-time simulation, Human Interface devices, Collision and rigid body dynamics, Game profiling. Introduction to Dx Studio, Introduction to Unity, Application layer, Game logic, Game views, managing memory, controlling the main loop, loading and caching game data, User Interface management, Game event management.</p> <p><b>Introduction To DirectX 11:</b> COM, Textures and Resources Formats, The swap chain and Page flipping, Depth Buffering, Texture Resource Views, Multisampling Theory and MS in Direct3D, Feature Levels</p> <p><b>Game Platform and Development:</b> 2D and 3D Game development using Python-Pygame, ModernGL, Numpy, PyGLM,Ursina (OpenGL)</p>	<b>15</b>
<b>III</b>	<p><b>Unity Development Environment:</b> IDE Basics, Unity Concepts, Sprites, Game Loops and Functions, Game Design Strategies.</p> <p><b>Simple Movement, Operations and Object Oriented Concepts:</b> Simple Rotation and Scaling, Rigidbody Components, Unity Colliders, Physics Materials, Scripting Collision Events, Primitive Data and Math, Decisions and Flow Control, Loops and Arrays, Game Design Strategies, Exceptions and Debugging, Defining Classes, Functions, Organizing and Managing Game Objects,</p> <p><b>Advanced Programming:</b> Virtual Worlds, Scrolling Games, Animation, Sound Effects, Advanced Game Physics, Multiple Scenes, Artificial Intelligence, User Interfaces, Game Art, Publishing Games.</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Mathematics for Computer Graphics, John Vince, Springer-Verlag London, 5th Edition,2017</li> <li>2. Mathematics for 3D Game Programming and Computer Graphic, Eric Lengyel, Delmar</li> <li>3. Introduction To 3D Game Programming With Directx® 11,Frank D Luna, Mercury Learning And Information,2012.</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Computer Graphics, C Version, Donald Hern and Pauline Baker, Pearson Education, 2nd Edition, 1997</li> <li>2. HLSL Development Cookbook, Doron Feinstein, PACKT Publishing,2013</li> <li>3. <a href="https://docs.unity3d.com/Manual/index.html">https://docs.unity3d.com/Manual/index.html</a></li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP5042</b>	<b>Game Programming – Practical</b>	<b>1</b>	<b>3</b>
1	Setup DirectX 11, Window Framework and Initialize Direct3D Device, Loading models into DirectX 11 and rendering		
2	Learn Basic Game Designing Techniques with pygame.		
3	Develop Snake Game using pygame		
4	Create 2D Target Shooting Game		
5	Creating 2D Infinite Scrolling Background		
6	Create Camera Shake Effect in Unity		
7	Design and Animate Game Character in Unity		
8	Create Snowfall Particle effect in Unity		
9	Develop Android Game with Unity		
10	Create Intelligent enemies in Unity		

Course Code	Course Title	Credits	Lectures /Week
USCS5051	Project Management	2	3
<p><b>About the Course:</b> The Project Management course emphasizes on the importance of effective project management in the field. Students will gain a solid understanding of project management principles, processes, and knowledge areas while exploring their application in computer science projects. The course covers key topics such as project selection and initiation, scope management, time management, cost management, quality management, human resource management, risk management, procurement management, integration management, and advanced topics relevant to the digital age. Through a combination of theory and practical exercises, students will develop the skills necessary to successfully plan, execute, and control projects in the dynamic and rapidly evolving field of Computer Science.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the fundamental concepts and characteristics of project management, including project selection, initiation, and project governance.</li> <li>• Develop skills in scope management, time management, cost management, quality management, and risk management to effectively plan and control projects.</li> <li>• Acquire knowledge of human resource management techniques, including team development, conflict resolution, and communication management.</li> <li>• Gain insight into agile project management methodologies and tools, enabling the successful management of iterative and incremental development.</li> <li>• Explore advanced topics such as stakeholder management, project leadership, project governance, and the role of technology in project management.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Apply project management principles, processes, and best practices to plan, execute, and control projects effectively.</li> <li>• Develop project charters, define project scopes, and create work breakdown structures (WBS) to establish project objectives and deliverables.</li> <li>• Create project schedules, estimate resource requirements, and monitor project progress using appropriate project management techniques.</li> <li>• Employ quality assurance and control measures to ensure project deliverables meet stakeholder expectations and industry standards.</li> <li>• Demonstrate effective leadership and teamwork skills, as well as the ability to manage stakeholders, resolve conflicts, and make ethical decisions in project management settings.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Project Management</b></p> <p><b>Introduction to Project Management:</b> Definition and characteristics of a project, Importance of project management, Project management processes and knowledge areas</p> <p><b>Project Selection, Initiation and scope Management:</b> Project selection criteria and methods, Project initiation and charter development, Stakeholder</p>	15	

	<p>identification and analysis, Scope planning and definition, Work Breakdown Structure (WBS) development, Scope verification and change control</p> <p><b>Project Time &amp; Cost Management:</b> Activity definition and sequencing, Estimating activity durations and resources, Developing the project schedule, Schedule control and monitoring, Cost estimation techniques, Budget development and monitoring, Earned Value Management (EVM), Cost control and analysis</p>	
<b>II</b>	<p><b>Project Execution and Control</b></p> <p><b>Project Quality &amp; Risk Management:</b> Quality planning and standards, Quality assurance and control, Process improvement and Six Sigma concept, Risk Management-Risk identification and assessment, Risk response planning, Risk monitoring and control</p> <p><b>Project Resource &amp; Procurement Management:</b> Team development and roles, Staffing, training, and motivation, Conflict resolution and communication management, Procurement planning and contract types, Solicitation, source selection, and contract administration, Vendor management and relationship building</p> <p><b>Project Integration Management:</b> Project integration processes and activities, Change management and project closure, Lessons learned and knowledge transfer</p>	<b>15</b>
<b>III</b>	<p><b>Advanced Topics in Project Management</b></p> <p><b>Agile Project &amp; Management in the Digital Age:</b> Agile principles and methodologies, Agile project planning and execution, Managing iterative and incremental development, Role of technology in project management, Virtual teams and distributed project management, Tools and software for project planning and collaboration</p> <p><b>Effective People Management in Projects:</b> Leadership styles and characteristics, Team building and motivation techniques, Emotional intelligence in project management, Stakeholder identification and analysis, Stakeholder engagement and communication strategies, Conflict resolution and negotiation skills</p> <p><b>Project Governance and Ethics:</b> Project governance structures and accountability, Ethical considerations in project management, Professional responsibility and codes of conduct</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Project Management for Business and Technology, 3rd edition, Pearson Education. John M. Nicholas, 2000</li> <li>2. Information Technology Project Management, by Jack T. Marchewka, 4th Wiley India 2013.</li> <li>3. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition 6th Edition, Project Management Institute, 2017</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Introduction to Software Project Management by Adolfo Villafiorita · 2016, CRC press, e book format.</li> <li>2. Project Management Professional Workbook, Claudia M. Baca, Patti M. Jansen, Sybex Publication, 2013</li> <li>3. Project Management, by S. J. Mantel, J. R. Meredith and etal., 1<sup>st</sup> edition, Wiley India, 2009.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCS5052	Operations Research	2	3
<p><b>About the Course:</b> This course introduces Operations Research (OR) and its application in decision-making. It covers the fundamental concepts, techniques, and methodologies used in OR, with a focus on linear programming and duality. Students will learn how to formulate and solve optimization problems, analyze sensitivity in linear programming, and explore advanced topics such as goal programming, transportation problems, and assignment problems. The course also introduces computer software commonly used in OR for problem solving.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To understand the fundamental principles and approaches of Operations Research.</li> <li>• To develop skills in formulating and solving linear programming problems.</li> <li>• To analyze the duality in linear programming and its managerial significance.</li> <li>• To apply sensitivity analysis techniques to assess the impact of changes in linear programming models.</li> <li>• To explore advanced topics in Operations Research, including goal programming, transportation problems, and assignment problems.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Define and explain the key concepts and features of Operations Research.</li> <li>• Formulate and solve linear programming models using appropriate techniques.</li> <li>• Apply duality concepts to analyze and interpret the results of linear programming problems.</li> <li>• Conduct sensitivity analysis to assess the robustness and flexibility of linear programming solutions.</li> <li>• Apply advanced techniques such as goal programming, transportation problems, and assignment problems to real-world decision-making scenarios.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Operations Research(OR)</b></p> <p><b>Exploring Operations Research</b> – A Quantitative Approach to Decision-Making, Definitions, Features, OR Approach to Problem Solving, Models and Modelling in Operations Research, Advantages of Model Building and Operations Research Study, Applications of Operations Research, Computer Software for Operations Research</p> <p><b>Linear Programming and Duality:</b> Linear Programming: Applications and Model Formulation, Structure of Linear Programming Model, General Structure of an LP Model, Assumptions of an LP Model, Advantages &amp; Limitations Linear Programming, Application Areas, General Mathematical Model of Linear Programming Problem, Examples of LP Model Formulation</p> <p><b>Linear Programming-The Graphical Method:</b> Important Definitions, Graphical Solution Methods of LP Problems, The Simplex Method: Introduction, Standard form of an LP Problem, Simplex Algorithm</p>	15	



	(Maximization Case), Simplex Algorithm (Minimization Case), Two-Phase Method, Big-M Method	
<b>II</b>	<p><b>Linear Programming</b></p> <p><b>Duality in Linear Programming:</b> Introduction, Formulation of Dual Linear Programming Problem, Symmetrical Form, Economic Interpretation of Dual Variables, Economic Interpretation of Dual Constraints, Rules for Constructing the Dual from Primal, Standard Results on Duality, Principle of Complementary Slackness, Managerial Significance of Duality</p> <p><b>Sensitivity Analysis in Linear Programming:</b> Introduction, Sensitivity Analysis</p> <p><b>Integer Linear Programming:</b> Introduction, Types of Integer Programming Problems, Enumeration and Cutting Plane Solution Concept, Branch and Bound Method, Applications of Zero-One Integer Programming</p>	<b>15</b>
<b>III</b>	<p><b>Advanced Topics in Operations Research</b></p> <p><b>Goal Programming:</b> Introduction, Difference Between LP and GP Approach, Concept of Goal Programming, Distinction among Objectives, Goals and Constraints, Goal Programming Model Formulation</p> <p><b>Transportation Problem:</b> Introduction, Mathematical Model of Transportation Problem, General Mathematical Model of Transportation Problem, The Transportation Algorithm, Methods for Finding Initial Solution</p> <p><b>Assignment Problem:</b> Introduction, Mathematical Models of Assignment Problem, Solution Methods of Assignment Problem, Hungarian Method for Solving Assignment Problem</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Operations Research: Theory and Applications, J K Sharma, Trinity Press, 6th Edition , 2017</li> <li>2. Introduction to Operations Research, Frederick S. Hillier, Gerald J. Lieberman, McGraw Hill Education; 11th edition, 2021</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Operations Research, P K Gupta, S. Chand Publications, 7th Edition, 2018</li> <li>2. Operations Research, P. Rama Murthy, New Age Publication, 2nd Edition</li> <li>3. Operations Research: An Introduction, 10th Edition, Hamdy A. Taha, Pearson Education, 2019</li> <li>4. Operations Research (Schaums Outline Series), Richard Bronson and Govindasami Naadimuthu, McGraw Hill Education, 2nd Edition, 2017</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP505</b>	<b>Project Work – I</b>	<b>2</b>	<b>3</b>
<b><u>Refer to the Project Guidelines at the end</u></b>			

## Semester VI

Course Code	Course Title	Credits	Lectures /Week
USCS601	Data Science	2	3
<p><b>About the Course:</b> This course introduces the field of Data Science, covering the fundamental concepts, techniques, and tools used for data analysis, machine learning, and data visualization. Students will learn how to preprocess and analyze data, build predictive models, evaluate model performance, and effectively communicate insights through visualizations. The course also explores data management principles and practices. Practical hands-on exercises and projects using popular Data Science libraries and technologies are included to reinforce the concepts learned.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the foundations and scope of Data Science, including its applications and comparison to related fields like Business Intelligence and Artificial Intelligence.</li> <li>• Develop skills in data preprocessing, including cleaning, transforming, selecting, and merging data, to ensure data quality and suitability for analysis.</li> <li>• Gain knowledge of machine learning algorithms and techniques, such as regression, classification, clustering, and ensemble learning, to build predictive models and make data-driven decisions.</li> <li>• Learn how to evaluate and select models using appropriate evaluation metrics and cross-validation techniques to ensure reliable and robust model performance.</li> <li>• Develop proficiency in data visualization techniques and tools to effectively communicate insights and tell compelling stories using data.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Apply data preprocessing techniques to clean and transform raw data, handle missing values and outliers, and merge datasets.</li> <li>• Implement machine-learning algorithms to perform tasks such as regression, classification, clustering, and ensemble learning.</li> <li>• Evaluate and compare different machine learning models using appropriate evaluation metrics and cross-validation techniques.</li> <li>• Create informative and visually appealing data visualizations to communicate insights and patterns in data.</li> <li>• Understand the principles and practices of data management, including data governance, data quality assurance, and data privacy considerations.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Data Science and Data Preprocessing</b></p> <p><b>What is Data Science?:</b> Definition and scope of Data Science, Applications and domains of Data Science, Comparison with other fields like Business Intelligence (BI), Artificial Intelligence (AI), Machine Learning (ML), and</p>	15	

	<p>Data Warehousing/Data Mining (DW-DM)</p> <p><b>Data Types and Sources:</b> Different types of data: structured, unstructured, semi-structured, Data sources: databases, files, APIs, web scraping, sensors, social media</p> <p><b>Data Preprocessing:</b> Data cleaning: handling missing values, outliers, duplicates, Data transformation: scaling, normalization, encoding categorical variables, Feature selection: selecting relevant features/columns, Data merging: combining multiple datasets</p> <p><b>Data Wrangling and Feature Engineering:</b> Data wrangling techniques: reshaping, pivoting, aggregating, Feature engineering: creating new features, handling time-series data Dummification: converting categorical variables into binary indicators, Feature scaling: standardization, normalization</p> <p><b>Tools and Libraries:</b> Introduction to popular libraries and technologies used in Data Science like Pandas, NumPy, Sci-kit Learn, etc.</p>	
<p style="text-align: center;"><b>II</b></p>	<p><b>Data Analysis and Machine Learning</b></p> <p><b>Exploratory Data Analysis (EDA):</b> Data visualization techniques: histograms, scatter plots, box plots, etc., Descriptive statistics: mean, median, mode, standard deviation, etc., Hypothesis testing: t-tests, chi-square tests, ANOVA, etc.</p> <p><b>Introduction to Machine Learning:</b> Supervised learning: classification and regression, Unsupervised learning: clustering and dimensionality reduction, Bias-variance tradeoff, underfitting, and overfitting</p> <p><b>Regression Analysis:</b> Simple linear regression, Multiple linear regression, Stepwise regression, Logistic regression for classification</p> <p><b>Model Evaluation and Selection:</b> Techniques for evaluating model performance: accuracy, precision, recall, F1-score, Confusion matrix and ROC curve analysis, Cross-validation: k-fold cross-validation, stratified cross-validation, Hyperparameter tuning and model selection</p> <p><b>Machine Learning Algorithms:</b> Decision Trees and Random Forests, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Ensemble Learning: Boosting and Bagging, K-Nearest Neighbors (K-NN), Gradient Descent for optimization</p>	<p style="text-align: center;"><b>15</b></p>
<p style="text-align: center;"><b>III</b></p>	<p><b>Model Evaluation, Data Visualization, and Management</b></p> <p><b>Model Evaluation Metrics:</b> Accuracy, precision, recall, F1-score, Area Under the Curve (AUC), Evaluating models for imbalanced datasets</p> <p><b>Data Visualization and Communication:</b> Principles of effective data visualization, Types of visualizations: bar charts, line charts, scatter plots, etc. Visualization tools: matplotlib, seaborn, Tableau, etc. Data storytelling: communicating insights through visualizations</p> <p><b>Data Management:</b> Introduction to data management activities, Data pipelines: data extraction, transformation, and loading (ETL), Data governance and data quality assurance, Data privacy and security considerations</p>	<p style="text-align: center;"><b>15</b></p>

**Textbook(s):**

1. Data Science from Scratch First Principles with Python- Joel Grus O'reilly, 2nd Edition
2. Advancing into Analytics From Excel to Python and R, George Mount, Oreilly, First Edition
3. Introduction to Machine Learning with Python, Andreas C. Muller, Sarah Guido, Oreilly, First Edition

**Additional Reference(s):**

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015
3. Hands-On Programming with R, Garrett Golemund,1st Edition, 2014
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP601</b>	<b>Data Science – Practical</b>	<b>1</b>	<b>3</b>
1	<p>Introduction to Excel</p> <ul style="list-style-type: none"> <li>• Perform conditional formatting on a dataset using various criteria.</li> <li>• Create a pivot table to analyze and summarize data.</li> <li>• Use VLOOKUP function to retrieve information from a different worksheet or table.</li> <li>• Perform what-if analysis using Goal Seek to determine input values for desired output.</li> </ul>		
2	<p>Data Frames and Basic Data Pre-processing</p> <ul style="list-style-type: none"> <li>• Read data from CSV and JSON files into a data frame.</li> <li>• Perform basic data pre-processing tasks such as handling missing values and outliers.</li> <li>• Manipulate and transform data using functions like filtering, sorting, and grouping.</li> </ul>		
3	<p>Feature Scaling and Dummification</p> <ul style="list-style-type: none"> <li>• Apply feature-scaling techniques like standardization and normalization to numerical features.</li> <li>• Perform feature dummification to convert categorical variables into numerical representations.</li> </ul>		
4	<p>Hypothesis Testing</p> <ul style="list-style-type: none"> <li>• Formulate null and alternative hypotheses for a given problem.</li> <li>• Conduct a hypothesis test using appropriate statistical tests (e.g., t-test, chi-square test).</li> <li>• Interpret the results and draw conclusions based on the test outcomes.</li> </ul>		
5	<p>ANOVA (Analysis of Variance)</p> <ul style="list-style-type: none"> <li>• Perform one-way ANOVA to compare means across multiple groups.</li> <li>• Conduct post-hoc tests to identify significant differences between group means.</li> </ul>		
6	<p>Regression and Its Types</p> <ul style="list-style-type: none"> <li>• Implement simple linear regression using a dataset.</li> <li>• Explore and interpret the regression model coefficients and goodness-of-fit measures.</li> <li>• Extend the analysis to multiple linear regression and assess the impact of additional predictors.</li> </ul>		
7	<p>Logistic Regression and Decision Tree</p> <ul style="list-style-type: none"> <li>• Build a logistic regression model to predict a binary outcome.</li> <li>• Evaluate the model's performance using classification metrics (e.g., accuracy, precision, recall).</li> <li>• Construct a decision tree model and interpret the decision rules for classification.</li> </ul>		

8	<p>K-Means Clustering</p> <ul style="list-style-type: none"> <li>• Apply the K-Means algorithm to group similar data points into clusters.</li> <li>• Determine the optimal number of clusters using elbow method or silhouette analysis.</li> <li>• Visualize the clustering results and analyze the cluster characteristics.</li> </ul>
9	<p>Principal Component Analysis (PCA)</p> <ul style="list-style-type: none"> <li>• Perform PCA on a dataset to reduce dimensionality.</li> <li>• Evaluate the explained variance and select the appropriate number of principal components.</li> <li>• Visualize the data in the reduced-dimensional space.</li> </ul>
10	<p>Data Visualization and Storytelling</p> <ul style="list-style-type: none"> <li>• Create meaningful visualizations using data visualization tools</li> <li>• Combine multiple visualizations to tell a compelling data story.</li> <li>• Present the findings and insights in a clear and concise manner.</li> </ul>

Course Code	Course Title	Credits	Lectures /Week
USCS602	Cloud Computing and Web Services	2	3
<p><b>About the Course:</b> The course "Cloud Computing and Web Services" provides an in-depth understanding of cloud computing fundamentals and web service technologies. Students will learn about different types of clouds, cloud deployment models, and cloud platforms. They will also explore key concepts of virtualization, security in cloud computing, and popular cloud computing platforms such as OpenStack and AWS. Through practical exercises and hands-on projects, students will gain the skills required to design, deploy, and manage cloud-based applications and services.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the basics of cloud computing, including types of clouds, deployment models, and essential characteristics of cloud platforms.</li> <li>• Explore web services technologies such as SOAP and REST and understand their role in distributed computing and parallel computing.</li> <li>• Gain proficiency in utilizing virtualization technologies, including creating virtual machines and managing virtualized environments using tools like KVM and oVirt.</li> <li>• Explore and utilize popular cloud computing platforms such as OpenStack and AWS to architect, deploy, and manage cloud-based applications and services.</li> <li>• Learn about cloud security fundamentals, including confidentiality, integrity, availability, and secure development practices.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Demonstrate a comprehensive understanding of cloud computing concepts, including different types of clouds and their characteristics.</li> <li>• Implement and utilize web service technologies, such as SOAP and REST, to develop distributed and parallel computing applications.</li> <li>• Design, deploy, and manage cloud-based applications and services using popular cloud computing platforms such as OpenStack and AWS.</li> <li>• Apply secure development practices and implement cloud security policies to ensure the confidentiality, integrity, and availability of cloud software solutions.</li> <li>• Utilize virtualization technologies to create and manage virtualized environments, considering the benefits and drawbacks of virtualization.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Cloud Computing Basics</b></p> <p><b>Web Services</b> – Distributed Computing, Parallel Computing, WSDL structure, SOAP- Structure of SOAP Message (In JAX-WS), SOAP Messaging Architecture, SOAP Header, Client-side SOAP Handler, REST-What is REST? HTTP methods, Java API for RESTful Web Services (JAX-RS)</p>	15	



	<b>Virtulization:-</b> Characteristics of Virtualized Environments.. Pros and Cons of Virtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment.	
<b>II</b>	<p><b>Introduction to Cloud Computing:</b></p> <p>Definition, Types of Clouds, Deployment of software solutions and web applications, Types of Cloud Platforms, Essential characteristics – On-demand self-service, Broad network access, Location independent resource pooling ,Rapid elasticity , Measured service, Comparing cloud providers with traditional IT service providers</p> <p><b>Cloud Computing Software Security fundamentals:</b> Cloud Information Security Objectives, Confidentiality, Integrity, Availability, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Secure Development practices, Approaches to Cloud Software Requirement Engineering, Cloud Security Policy Implementation.</p>	<b>15</b>
<b>III</b>	<p><b>Cloud Applications</b></p> <p><b>CloudSim:</b> Introduction to Simulator, understanding CloudSim simulator, CloudSim Architecture(User code, CloudSim, GridSim, SimJava) Understanding Working platform for CloudSim,</p> <p><b>OpenStack:</b> Introduction to OpenStack, OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat</p> <p><b>AWS:</b> Architecting on AWS, Building complex solutions with Amazon Virtual Private Cloud (Amazon VPC)</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Java Web Services Up and Running 2nd edition, Martin Kalin, O'Reilly (2013)</li> <li>2. Pro Power Shell for Amazon Web Services, Brian Beach, Apress, 2014</li> <li>3. Enterprise Cloud Computing Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010</li> <li>4. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013</li> <li>5. OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. OpenStack Essentials, Dan Radez, PACKT Publishing, 2015 2</li> <li>2. OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O'Reilly Media, Inc., 2014</li> <li>3. <a href="https://www.openstack.org">https://www.openstack.org</a></li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP602</b>	<b>Cloud Computing and Web Services – Practical</b>	<b>1</b>	<b>3</b>
1	Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVA and .NET		
2	Create a Simple SOAP service.		
3	Create a Simple REST Service.		
4	Develop application to consume Google’s search / Google’s Map RESTful Web service.		
5	Installation and Configuration of virtualization using KVM.		
6	Develop application to download image/video from server or upload image/video to server using MTOM techniques		
7	Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Storage		
8	Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS),		
9	Using AWS Flow Framework develop application that includes a simple workflow. Workflow calls an activity to print hello world to the console. It must define the basic usage of AWS Flow Framework, including defining contracts, implementation of activities and workflow coordination logic and worker programs to host them		
10	Implementation of Openstack with user and private network creation.		

Course Code	Course Title	Credits	Lectures /Week
USCS6031	Wireless Sensor Networks	2	3
<p><b>About the Course:</b> This course provides a comprehensive understanding of Wireless Sensor Networks (WSNs) and their applications. It covers the fundamental concepts, architectural elements, advantages, and challenges of WSNs. Students will explore sensor node technology, network architecture, optimization goals, and design principles for WSNs. The course also delves into wireless transmission, telecommunication systems, and introduces the concepts of WSN operating systems and ad-hoc networks. Through practical examples and case studies, students will gain hands-on experience in medium access control protocols, routing strategies, transport control protocols, and WSN middleware architecture.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Provide students with a comprehensive understanding of Wireless Sensor Networks (WSNs), including their basic architectural elements, advantages, and challenges.</li> <li>• Introduce students to the key technologies and protocols used in WSNs, such as medium access control (MAC) protocols, routing strategies, and transport control protocols.</li> <li>• Familiarize students with wireless transmission principles and telecommunication systems relevant to WSNs, including frequency, signals, antennas, and satellite systems.</li> <li>• Develop students' practical skills in designing and implementing WSN solutions by exploring WSN operating systems, ad-hoc networks, and optimization goals.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Understand the fundamental concepts, architectural elements, and optimization goals of Wireless Sensor Networks (WSNs) and apply this knowledge to analyze and design WSN solutions.</li> <li>• Evaluate and compare different medium access control protocols and routing strategies in WSNs, and make informed decisions to ensure efficient and reliable communication.</li> <li>• Demonstrate knowledge of wireless transmission technologies, such as frequency, signals, antennas, and propagation, and analyze their impact on WSN performance.</li> <li>• Assess the role of telecommunication systems, satellite, broadcast systems in WSNs, and understand their applications and implications for WSN deployments.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction and Overview of Wireless Sensor Networks:</b> Basic Sensor Network Architectural Elements, Advantage and challenges, Applications, Sensor Node Technology, Sensor Taxonomy, WN Operating Environment, Radio Technology, Network architecture, Optimization goals and figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts.</p> <p><b>Wireless Sensor Network Operating Systems and Ad-hoc Networks:</b> Overview of Wireless Sensor Network Operating Systems, Examples of WSN Operating Systems Ad-hoc Networks in Wireless Sensor Networks, Characteristics and Challenges of Ad-hoc Networks in WSNs, Energy</p>	15	

	Efficiency Considerations in Ad-hoc Networks, Security and Privacy in Ad-hoc Networks, Examples of WSN OS, Ad-hoc Network.	
<b>II</b>	<p><b>Medium Access Control Protocol:</b> Fundamentals of MAC Protocols, Sensor-MAC Case Study</p> <p><b>Routing in WSN:</b> Routing Challenges and Design Issues in Wireless Sensor Networks, , IEEE 802.15.4 LR-WPANs Standard Case Study, Routing Strategies in Wireless Sensor Networks,</p> <p><b>Transport Control Protocol:</b> Traditional Transport Control Protocols, Transport Protocol Design Issues, WSN Middleware Architecture</p>	<b>15</b>
<b>III</b>	<p><b>Wireless Transmission:</b> Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems.</p> <p><b>Telecommunication, Satellite and Broadcast Systems:</b> Satellite and Broadcast Systems: GSM: Mobile services, System architecture, Radio interface, Protocols, Localization And Calling, Handover, security, New data services; DECT: System architecture, Protocol architecture; ETRA, UMTS and IMT- 2000. Satellite Systems: History, Applications, Basics: GEO, LEO, MEO; Routing, Localization, Handover.</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley &amp; Sons, 2017</li> <li>2. Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2015</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Fundamentals of Wireless Sensor Networks, Theory and Practice, Walteneus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011</li> <li>2. Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP6031</b>	<b>Wireless Sensor Networks – Practical</b>	<b>1</b>	<b>3</b>
1	Understanding the Sensor Node Hardware. (For Eg. Sensors, Nodes(Sensor mote), Base Station, Graphical User Interface.)		
2	Exploring and understanding TinyOS computational concepts: - Events, Commands and Task. - nesC model - nesC Components		
3	Understanding TOSSIM for - Mote-mote radio communication - Mote-PC serial communication		
4	Create and simulate a simple adhoc network		
5	Understanding, Reading and Analyzing Routing Table of a network.		
6	Create a basic MANET implementation simulation for Packet animation and Packet Trace		
7	Implement a Wireless sensor network simulation.		
8	Create MAC protocol simulation implementation for wireless sensor Network.		
9	Simulate Mobile Adhoc Network with Directional Antenna		
10	Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them		

Course Code	Course Title	Credits	Lectures /Week
USCS6032	Information Retrieval	2	3
<p><b>About the Course:</b> This course introduces the principles, techniques, and technologies underlying information retrieval (IR) systems. Students will explore the fundamental concepts of document indexing, storage, and retrieval, as well as advanced topics such as retrieval models, text categorization, web information retrieval, and evaluation techniques. Through a combination of theoretical study, practical exercises, and reference to industry-standard books, students will gain a solid foundation in the field of information retrieval.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To understand the fundamental principles and components of information retrieval systems.</li> <li>• To explore various techniques for document indexing, storage, and retrieval.</li> <li>• To analyze and compare different retrieval models and understand their strengths and limitations.</li> <li>• To gain practical experience in implementing and evaluating information retrieval systems.</li> <li>• To explore advanced topics in information retrieval, such as web information retrieval and machine learning techniques.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Explain the key components and principles of information retrieval systems.</li> <li>• Apply indexing, storage, and retrieval techniques to efficiently retrieve relevant documents.</li> <li>• Compare and contrast different retrieval models and select appropriate models for specific search scenarios.</li> <li>• Develop practical skills in implementing and evaluating information retrieval systems.</li> <li>• Demonstrate an understanding of advanced topics in information retrieval, including web search and machine learning techniques.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Foundations of Information Retrieval</b></p> <p><b>Introduction to Information Retrieval (IR) systems:</b> Definition and goals of information retrieval, Components of an IR system, Challenges and applications of IR</p> <p><b>Document Indexing, Storage, and Compression:</b> Inverted index construction and compression techniques, Document representation and term weighting, Storage and retrieval of indexed documents,</p> <p><b>Retrieval Models:</b> Boolean model: Boolean operators, query processing, Vector space model: TF-IDF, cosine similarity, query-document matching, Probabilistic model: Bayesian retrieval, relevance feedback</p> <p><b>Spelling Correction in IR Systems:</b> Challenges of spelling errors in queries and documents, Edit distance and string similarity measures, Techniques for</p>	15	

	<p>spelling correction in IR systems</p> <p><b>Performance Evaluation:</b> Evaluation metrics: precision, recall, F-measure, average precision, Test collections and relevance judgments, Experimental design and significance testing</p>	
<b>II</b>	<p><b>Advanced Topics in Information Retrieval</b></p> <p><b>Text Categorization and Filtering:</b> Text classification algorithms: Naive Bayes, Support Vector Machines, Feature selection and dimensionality reduction, Applications of text categorization and filtering</p> <p><b>Text Clustering for Information Retrieval:</b> Clustering techniques: K-means, hierarchical clustering, Evaluation of clustering results, Clustering for query expansion and result grouping</p> <p><b>Web Information Retrieval:</b> Web search architecture and challenges, Crawling and indexing web pages, Link analysis and PageRank algorithm</p> <p><b>Learning to Rank:</b> Algorithms and Techniques, Supervised learning for ranking: RankSVM, RankBoost, Pairwise and listwise learning to rank approaches Evaluation metrics for learning to rank</p> <p><b>Link Analysis and its Role in IR Systems:</b> Web graph representation and link analysis algorithms, HITS and PageRank algorithms, Applications of link analysis in IR systems</p>	<b>15</b>
<b>III</b>	<p><b>Advanced Topics in Information Retrieval</b></p> <p><b>Crawling and Near-Duplicate Page Detection:</b> Web page crawling techniques: breadth-first, depth-first, focused crawling, Near-duplicate page detection algorithms, Handling dynamic web content during crawling</p> <p><b>Advanced Topics in IR:</b> Text Summarization: extractive and abstractive methods, Question Answering: approaches for finding precise answers, Recommender Systems: collaborative filtering, content-based filtering</p> <p><b>Cross-Lingual and Multilingual Retrieval:</b> Challenges and techniques for cross-lingual retrieval, Machine translation for IR, Multilingual document representations and query translation, Evaluation Techniques for IR Systems</p> <p><b>User-based evaluation:</b> user studies, surveys, Test collections and benchmarking, Online evaluation methods: A/B testing, interleaving experiments</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books</li> <li>2. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Ricci, F, Rokach, L. Shapira, B. Kantor, —Recommender Systems Handbook, First Edition.</li> <li>2. Bruce Croft, Donald Metzler, and Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson Education.</li> <li>3. Stefan Butcher, Charlie Clarke, Gordon Cormack, Information Retrieval: Implementing and Evaluating Search Engines, MIT Press.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP6032</b>	<b>Information Retrieval – Practical</b>	<b>1</b>	<b>3</b>
1	Document Indexing and Retrieval <ul style="list-style-type: none"> <li>• Implement an inverted index construction algorithm.</li> <li>• Build a simple document retrieval system using the constructed index.</li> </ul>		
2	Retrieval Models <ul style="list-style-type: none"> <li>• Implement the Boolean retrieval model and process queries.</li> <li>• Implement the vector space model with TF-IDF weighting and cosine similarity.</li> </ul>		
3	Spelling Correction in IR Systems <ul style="list-style-type: none"> <li>• Develop a spelling correction module using edit distance algorithms.</li> <li>• Integrate the spelling correction module into an information retrieval system.</li> </ul>		
4	Evaluation Metrics for IR Systems <ul style="list-style-type: none"> <li>• Calculate precision, recall, and F-measure for a given set of retrieval results.</li> <li>• Use an evaluation toolkit to measure average precision and other evaluation metrics.</li> </ul>		
5	Text Categorization <ul style="list-style-type: none"> <li>• Implement a text classification algorithm (e.g., Naive Bayes or Support Vector Machines).</li> <li>• Train the classifier on a labelled dataset and evaluate its performance.</li> </ul>		
6	Clustering for Information Retrieval <ul style="list-style-type: none"> <li>• Implement a clustering algorithm (e.g., K-means or hierarchical clustering).</li> <li>• Apply the clustering algorithm to a set of documents and evaluate the clustering results.</li> </ul>		
7	Web Crawling and Indexing <ul style="list-style-type: none"> <li>• Develop a web crawler to fetch and index web pages.</li> <li>• Handle challenges such as robots.txt, dynamic content, and crawling delays.</li> </ul>		
8	Link Analysis and PageRank <ul style="list-style-type: none"> <li>• Implement the PageRank algorithm to rank web pages based on link analysis.</li> <li>• Apply the PageRank algorithm to a small web graph and analyze the results.</li> </ul>		
9	Learning to Rank <ul style="list-style-type: none"> <li>• Implement a learning to rank algorithm (e.g., RankSVM or RankBoost).</li> <li>• Train the ranking model using labelled data and evaluate its effectiveness.</li> </ul>		
10	Advanced Topics in Information Retrieval <ul style="list-style-type: none"> <li>• Implement a text summarization algorithm (e.g., extractive or abstractive).</li> <li>• Build a question-answering system using techniques such as information extraction</li> </ul>		



Course Code	Course Title	Credits	Lectures /Week
USCS6041	Data Mining & Warehousing	2	3
<p><b>About the Course:</b> The course covers data warehousing, data mining, association rule mining, classification and prediction, and clustering. Students will learn about OLAP and OLTP, multi-dimensional data models, measures, concept of hierarchy, and data warehouse architecture. They will also explore different data mining functionalities. The course covers classification methods, prediction techniques, and classifier accuracy assessment. Students will learn various clustering methods and their applications to different data types, such as time-series data, text databases, and web mining.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the concept and framework of data warehousing and differentiate between OLAP and OLTP.</li> <li>• Gain knowledge of data mining techniques and their applications in knowledge discovery.</li> <li>• Acquire skills in data preprocessing, including handling missing data, cleaning, integration, and transformation.</li> <li>• Apply association rules mining algorithms such as APRIORI and FP-Growth to discover frequent item sets.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Explain the purpose and components of a data warehouse and differentiate it from transactional databases.</li> <li>• Perform OLAP operations on a multidimensional data model to analyze and query data.</li> <li>• Implement data preprocessing techniques to address missing data and prepare the data for mining.</li> <li>• Apply association rules mining algorithms to discover patterns and relationships in large datasets.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Data Warehouse:</b> Introduction, Necessity, Framework of the data warehouse, options, developing data warehouses, Differences between OLAP and OLTP, OLAP Operations in the Multidimensional Data Model, Back-End Tools and Utilities, Metadata Repository, Types of OLAP servers.</p> <p><b>DW Design Consideration And Dimensional Modeling:</b> Defining Dimensional Model, Granularity of Facts, Additivity of Facts, Functional dependency of the Data, Helper Tables, Implementation many-to-many relationships between fact and dimensional modeling.</p> <p><b>Data Warehouse Models:</b> Enterprise Data Warehouse (EDW), Data Mart, Virtual Data Warehouse, Hybrid Data Warehouse.</p>	15	
II	<p><b>Data Mining:</b> Introduction to Data Mining, Definition, Knowledge Discovery in Data (KDD), Kinds of databases, Data to be mined, Basic mining techniques, Data Mining Issues, Data Mining Metrics, Social Implications of Data Mining, Overview of Applications of Data Mining.</p>	15	

	<p><b>Data Preprocessing:</b> Data Processing prerequisites, Attributes and Data types, Statistical descriptions of data, Distance and similarity measures, Need for Preprocessing, Handling Missing data, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.</p> <p><b>Association Rules Mining:</b> Problem Definition, Frequent item set generation, The APRIORI Principle, Support and confidence measures, Association rule generation: APRIORI algorithm, FP-Growth Algorithms, Compact Representation of Frequent item Set: Maximal Frequent item set, closed frequent item set.</p>	
<b>III</b>	<p><b>Classification And Prediction:</b> Definition of classification, Model construction, Model Usage, Choosing algorithm, Decision tree Induction, Information gain, gain ratio, gini index, Bayesian Classification, Bayes Theorem, Naïve Bayes classifier, Linear Regression, Non-linear Regression, Logistic Regression.</p> <p><b>Validating Model:</b> Measuring performance of classifiers, Precision, Recall, F-measure, confusion matrix, cross-validation, Bootstrap.</p> <p><b>Clustering:</b> Types of data, Categorization of major clustering methods, K-means partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based clustering methods, Outlier analysis, Mining Time-Series and Sequence Data, Mining Text Databases, Mining the World Wide Web.</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Data Warehousing : Design, Development And Best Practices by Souendra Mohanty (Author), Tata McGraw Hill Education (Publisher).</li> <li>2. Jiawei Han, Michelin Kamber, “Data Mining-Concepts and techniques”, Morgan Kaufmann Publishers, Elsevier, 3rd Edition.</li> <li>3. Alex Berson, Stephen J.Smith, “Data warehousing Data mining and OLAP”, Tata McGraw- Hill, 2nd Edition.</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Arum K Pujari, “Data Mining Techniques”, 3rd Edition, Universities Press, 2005</li> <li>2. PualrajPonnaiah, Wiley, “Data Warehousing Fundamentals”, Student Edition, 2004.</li> <li>3. Ralph Kimball, Wiley, “The Data warehouse Life Cycle Toolkit”, Student Edition, 2006.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
<b>USCSP6041</b>	<b>Data Mining &amp; Warehousing – Practical</b>	<b>1</b>	<b>3</b>
1	Perform different operations of extraction, transformation, and loading (ETL) processes on a sample dataset using PowerBI.		
2	Integrate data from multiple sources by merging and transforming datasets using Python's pandas library and data manipulation techniques.		
3	Apply feature selection techniques like variance thresholding and correlation analysis using Python's <b>scikit-learn library</b> to reduce dimensionality in a dataset.		
4	<b>Discretize</b> continuous variables and create concept hierarchies for categorical variables in a market basket dataset using Python's pandas library.		
5	Implement the <b>Apriori algorithm</b> in Python to mine frequent itemsets from a retail transaction dataset and extract association rules.		
6	Build a <b>decision tree classifier</b> using Python's scikit-learn library to predict customer churn based on historical data.		
7	Implement a <b>Naive Bayes classifier</b> in Python using scikit-learn to classify emails as spam or non-spam based on their content.		
8	Implement a <b>linear regression</b> method to make predictions based on the sample data set using Python.		
9	Implement a <b>logistic regression</b> method to make predictions based on the sample data set using Python.		
10	Implement <b>K-means clustering</b> algorithm in Python using scikit-learn to group customers based on their purchasing behavior.		

Course Code	Course Title	Credits	Lectures /Week
USCS6042	Ethical Hacking	2	3
<p><b>About the Course:</b> This course provides an in-depth exploration of ethical hacking and penetration testing methodologies. Students will learn about hacking technology types, the phases of ethical hacking, footprinting, social engineering, system hacking, web server and application vulnerabilities, wireless hacking, and more. The course emphasizes hands-on lab exercises and real-world scenarios to develop practical skills in identifying and mitigating security vulnerabilities.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand the terminology and concepts related to ethical hacking and penetration testing.</li> <li>• Explore various hacking technologies and the skills required to become an ethical hacker.</li> <li>• Learn the different phases involved in ethical hacking and the methodologies used in penetration testing.</li> <li>• Gain knowledge of common hacking techniques, such as footprinting, scanning, enumeration, and session hijacking.</li> <li>• Develop proficiency in identifying and exploiting vulnerabilities in web servers, web applications, and wireless networks.</li> </ul>			
<p><b>Learning Outcomes:</b> After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Apply ethical hacking methodologies to conduct comprehensive security assessments and penetration tests.</li> <li>• Perform effective footprinting and reconnaissance techniques to gather critical information about target systems.</li> <li>• Identify and exploit vulnerabilities in various network and system components using appropriate tools and techniques.</li> <li>• Evaluate the security posture of web servers, web applications, and wireless networks, and recommend appropriate countermeasures.</li> <li>• Demonstrate an understanding of ethical and legal considerations in conducting ethical hacking activities and adhere to professional codes of conduct.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction:</b> Terminology, Hacking Technology Types, Ethical Hacking Phases, Hacktivism, Hacker Classes, Skills Required for an Ethical Hacker, Vulnerability Research, Ways to Conduct Ethical Hacking</p> <p><b>Footprinting:</b> Definition, Information Gathering Methodology, Competitive Intelligence, DNS Enumeration, Whois and ARIN Lookups, Types of DNS Records, Traceroute in Footprinting, E-Mail Tracking</p> <p><b>Social Engineering:</b> Common Types Of Attacks</p> <p><b>Scanning and Enumeration:</b> Port Scanning, Network Scanning, Vulnerability Scanning, CEH Scanning Methodology, Ping Sweep Techniques, Nmap Command Switches, SYN, Stealth, XMAS, NULL,</p>	15	

	IDLE, FIN Scans, Anonymizers, HTTP Tunneling Techniques, IP Spoofing Techniques, SNMP Enumeration, Steps Involved in Enumeration	
<b>II</b>	<p><b>System Hacking:</b> Password-Cracking Techniques, Types of Passwords, Keyloggers and Other Spyware Technologies, Escalating Privileges, Rootkits</p> <p><b>Sniffers:</b> Protocols Susceptible to Sniffing, Active and Passive Sniffing, ARP Poisoning, MAC Flooding, DNS Spoofing Techniques, Sniffing Countermeasures</p> <p><b>Denial of Service:</b> Types of DoS Attacks, Working of DoS Attacks, BOTs/BOTNETs, “Smurf” Attack, “SYN” Flooding, DoS/DDoS Countermeasures</p> <p><b>Session Hijacking:</b> Spoofing vs. Hijacking, Types, Sequence Prediction, Steps, Prevention</p> <p><b>Hacking Web Servers:</b> Web Server Vulnerabilities, Attacks against Web Servers, Patch Management Techniques, Web Server Hardening</p>	<b>15</b>
<b>III</b>	<p><b>Web Application Vulnerabilities:</b> Web Application Hacking, Web Application Threats, Google Hacking, Countermeasures</p> <p><b>Web-Based Password Cracking Techniques:</b> Authentication Types, Password Crackers, Countermeasures</p> <p><b>SQL Injection:</b> Steps, SQL Server Vulnerabilities, Countermeasures</p> <p><b>Buffer Overflows:</b> Types, Stack-Based Buffer Overflows, Mutation Techniques</p> <p><b>Wireless Hacking:</b> WEP, WPA Authentication Mechanisms, and Cracking Techniques, Wireless Sniffers, Rogue Access Points, Wireless Hacking Techniques, Securing Wireless Networks</p> <p><b>Penetration Testing Methodologies:</b> Methodologies, Steps, Automated Tools, Pen-Test Deliverables</p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. CEH official Certified Ethical Hacking Review Guide, Wiley India Edition</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Certified Ethical Hacker: Michael Gregg, Pearson Education</li> <li>2. Certified Ethical Hacker: Matt Walker, TMH.</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCSP6042	Ethical Hacking - Practical	1	3
1	<p>Google and Whois Reconnaissance</p> <ul style="list-style-type: none"> <li>• Use Google search techniques to gather information about a specific target or organization.</li> <li>• Utilize advanced search operators to refine search results and access hidden information.</li> <li>• Perform Whois lookups to retrieve domain registration information and gather details about the target's infrastructure.</li> </ul>		
2	<p>Password Encryption and Cracking with CrypTool and Cain and Abel</p> <ul style="list-style-type: none"> <li>• Password Encryption and Decryption: <ul style="list-style-type: none"> <li>○ Use CrypTool to encrypt passwords using the RC4 algorithm.</li> <li>○ Decrypt the encrypted passwords and verify the original values.</li> </ul> </li> <li>• Password Cracking and Wireless Network Password Decoding: <ul style="list-style-type: none"> <li>○ Use Cain and Abel to perform a dictionary attack on Windows account passwords.</li> <li>○ Decode wireless network passwords using Cain and Abel's capabilities.</li> </ul> </li> </ul>		
3	<p>Linux Network Analysis and ARP Poisoning</p> <ul style="list-style-type: none"> <li>• Linux Network Analysis: <ul style="list-style-type: none"> <li>○ Execute the ifconfig command to retrieve network interface information.</li> <li>○ Use the ping command to test network connectivity and analyze the output.</li> <li>○ Analyze the netstat command output to view active network connections.</li> <li>○ Perform a traceroute to trace the route packets take to reach a target host.</li> </ul> </li> <li>• ARP Poisoning: <ul style="list-style-type: none"> <li>○ Use ARP poisoning techniques to redirect network traffic on a Windows system.</li> <li>○ Analyze the effects of ARP poisoning on network communication and security.</li> </ul> </li> </ul>		
4	<p>Port Scanning with NMap</p> <ul style="list-style-type: none"> <li>• Use NMap to perform an ACK scan to determine if a port is filtered, unfiltered, or open.</li> <li>• Perform SYN, FIN, NULL, and XMAS scans to identify open ports and their characteristics.</li> <li>• Analyze the scan results to gather information about the target system's network services.</li> </ul>		
5	<p>Network Traffic Capture and DoS Attack with Wireshark and Nemesy</p> <ul style="list-style-type: none"> <li>• Network Traffic Capture: <ul style="list-style-type: none"> <li>○ Use Wireshark to capture network traffic on a specific network interface.</li> <li>○ Analyze the captured packets to extract relevant information and identify potential security issues.</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>• Denial of Service (DoS) Attack: <ul style="list-style-type: none"> <li>○ Use Nemesy to launch a DoS attack against a target system or network.</li> <li>○ Observe the impact of the attack on the target's availability and performance.</li> </ul> </li> </ul>
6	<p>Persistent Cross-Site Scripting Attack</p> <ul style="list-style-type: none"> <li>• Set up a vulnerable web application that is susceptible to persistent XSS attacks.</li> <li>• Craft a malicious script to exploit the XSS vulnerability and execute arbitrary code.</li> <li>• Observe the consequences of the attack and understand the potential risks associated with XSS vulnerabilities.</li> </ul>
7	<p>Session Impersonation with Firefox and Tamper Data</p> <ul style="list-style-type: none"> <li>• Install and configure the Tamper Data add-on in Firefox.</li> <li>• Intercept and modify HTTP requests to impersonate a user's session.</li> <li>• Understand the impact of session impersonation and the importance of session management.</li> </ul>
8	<p>SQL Injection Attack</p> <ul style="list-style-type: none"> <li>• Identify a web application vulnerable to SQL injection.</li> <li>• Craft and execute SQL injection queries to exploit the vulnerability.</li> <li>• Extract sensitive information or manipulate the database through the SQL injection attack.</li> </ul>
9	<p>Creating a Keylogger with Python</p> <ul style="list-style-type: none"> <li>• Write a Python script that captures and logs keystrokes from a target system.</li> <li>• Execute the keylogger script and observe the logged keystrokes.</li> <li>• Understand the potential security risks associated with keyloggers and the importance of protecting against them.</li> </ul>
10	<p>Exploiting with Metasploit (Kali Linux)</p> <ul style="list-style-type: none"> <li>• Identify a vulnerable system and exploit it using Metasploit modules.</li> <li>• Gain unauthorized access to the target system and execute commands or extract information.</li> <li>• Understand the ethical considerations and legal implications of using Metasploit for penetration testing.</li> </ul>

Course Code	Course Title	Credits	Lectures /Week
USCS6051	Customer Relationship Management	2	3
<p><b>About the Course:</b> This course on Customer Relationship Management (CRM) provides an in-depth understanding of the principles, strategies, and tools necessary for managing customer relationships effectively. Students will explore the various forms of CRM and its impact on business performance. Additionally, the course covers customer acquisition, retention, and the measurement of customer-perceived value. Students will also gain insights into strategic and operational CRM, including customer portfolio management, marketing automation, and service automation. The course concludes with an examination of analytical CRM and the implementation of CRM strategies through real-life case studies.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• To provide students with a comprehensive understanding of CRM concepts, theories, and models.</li> <li>• To equip students with the knowledge and skills to manage the customer journey, including customer acquisition and retention.</li> <li>• To explore the factors that contribute to customer-perceived value and the role of CRM in enhancing the customer experience.</li> <li>• To familiarize students with strategic and operational CRM approaches, including customer portfolio management and marketing automation.</li> <li>• To introduce students to the analytical aspects of CRM, including data management, analytics for strategy and tactics, and the implementation of CRM systems.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Students will be able to define and explain the various forms of CRM and their relevance to business contexts.</li> <li>• Students will acquire the skills to manage the customer journey effectively, including implementing customer acquisition and retention programs.</li> <li>• Students will understand the importance of customer-perceived value and its impact on customer satisfaction, loyalty, and business performance.</li> <li>• Students will be able to apply strategic and operational CRM approaches, such as customer portfolio management and marketing automation, to enhance organizational effectiveness.</li> <li>• Students will develop proficiency in analytical CRM techniques, including data management, analytics for strategy and tactics, and the successful implementation of CRM systems. They will also be able to analyze and draw insights from real-life case studies and success stories related to CRM.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Understanding Customer Relationships</b></p> <p><b>Introduction to CRM:</b> Three forms of CRM, The changing face of CRM, Misunderstandings about CRM, Defining CRM, CRM constituencies, Commercial contexts of CRM, Models of CRM, Understanding relationships, Relationship Quality, Customer life-time value, Customer</p>	15	



	<p>satisfaction, loyalty and business performance, Relationship management theories, Benefits of CRM</p> <p><b>Managing the customer journey:</b> customer acquisition, Portfolio purchasing, Prospecting, Key performance indicators of customer acquisition programs, Operational CRM tools that help customer acquisition, Customer retention, Economics and Strategies of customer retention, Key performance indicators of customer retention programs.</p> <p><b>Managing customer-experienced value:</b> Understanding value, modeling customer-perceived value, its sources, Customization, Value through the marketing mix, Customer Experience concepts, Service marketing, Total quality management, relationship management, CRM's influence on CX, How CRM software applications influence customer experience</p>	
<b>II</b>	<p><b>Strategic and Operational CRM</b></p> <p><b>Customer portfolio management:</b> Portfolio, customer, Basic disciplines for CPM, CPM in the business-to-business context, customer portfolio management tools, strategically significant customers, The seven core customer management strategies</p> <p><b>Marketing automation:</b> Introduction to marketing automation, Benefits, Software applications for marketing, Sales force automation</p> <p><b>Service automation:</b> Introduction, customer service, Modeling service quality, Customer service standards, service automation, Benefits, Software applications for service</p>	<b>15</b>
<b>III</b>	<p><b>Analytical CRM</b></p> <p><b>Developing and managing customer-related databases:</b> Corporate customer-related data, Structured and unstructured data, Developing a customer-related database, Data integration, Data warehousing, Data marts, Analytics for CRM strategy and tactics, Big data analytics, Analytics for structured data, ways to generate analytical insight, Data-mining procedures, Artificial intelligence (AI), machine learning (ML) and deep learning (DL)</p> <p><b>Implementing CRM:</b> Introduction, develop the CRM strategy, build CRM project foundations, needs specification and partner selection, project implementation, performance evaluation</p> <p><b>Case studies and success stories related to CRM</b></p>	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Customer Relationship Management Concepts and Technologies, Francis Buttle, Stan Maklan, Routledge Taylor and Francis Group, 2019</li> <li>2. Jagdish N Sheth, Parvatiyar Atul, G Shainesh, Customer Relationship Management- Emerging Concepts, Tools and Applications, 2017</li> <li>3. Anderrson Kristin , Carol Kerr, Customer Relationship Management, Tata McGraw-Hill, 2017</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. V. Kumar &amp; Werner J., CUSTOMER RELATIONSHIP MANAGEMENT, Willey India, 2008</li> <li>2. S. Shanmugasundaram, CUSTOMER RELATIONSHIP MANAGEMENT, Prentice Hall of India Private Limited, New Delhi, 2008</li> </ol>		

Course Code	Course Title	Credits	Lectures /Week
USCS6052	Cyber Laws and IPR	2	3
<p><b>About the Course:</b> The course on Cyber Laws and IPR provides a comprehensive understanding of the legal aspects and regulations related to cyberspace and information technology. The course covers a wide range of topics, including basic concepts, internet technology, network security, cyber law, e-commerce, electronic signatures, cyber crimes, privacy, intellectual property rights, and more. Students will explore the legal framework governing cyberspace and develop an understanding of the legal and ethical issues associated with information technology.</p>			
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand fundamental concepts of cyber laws and their relevance in the digital age.</li> <li>• Examine legal frameworks and regulations in cyber laws, including the Information Technology Act 2000 in India.</li> <li>• Explore key issues in cyber laws such as e-commerce, e-governance, and electronic records and contracts.</li> <li>• Gain knowledge of cybercrimes, enforcement mechanisms, and the role of the Cyber Appellate Tribunal.</li> <li>• Analyze emerging issues in cyber laws, including liability of ISPs, privacy concerns, and jurisdictional aspects.</li> <li>• Understand intellectual property rights and online regulations, including copyrights, patents, and domain name disputes.</li> </ul>			
<p><b>Learning Outcomes:</b></p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li>• Demonstrate a comprehensive understanding of cyber laws and their application in the digital age.</li> <li>• Evaluate legal frameworks and regulations governing cyber laws.</li> <li>• Identify and assess key issues in cyber laws, such as e-commerce, e-governance, and electronic records and contracts.</li> <li>• Understand cyber crimes, enforcement mechanisms, and the role of the Cyber Appellate Tribunal.</li> <li>• Analyze emerging issues in cyber laws, including liability of ISPs, privacy concerns, and jurisdictional complexities.</li> <li>• Recognize intellectual property rights and online regulations, including copyrights, patents, and domain name disputes.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Cyber Laws and Technology:</b> Basic Concepts, Internet and Advantages and Disadvantages of Internet Technology, Network and Network Security</p> <p><b>Legal Framework and Regulations:</b> Cyber Law &amp; Components of Cyber Law, Cyber Law in India: An Overview of Information Technology Act 2000, Cryptography, Encryption Technique &amp; Algorithm and Digital</p>	15	

	Signature & Electronic Signature <b>Key Issues in Cyber Laws:</b> E-Commerce, E-Governance, E-Record & E-Contract, Regulator, Certifying Authority, Electronic Signature Certificates	
<b>II</b>	<b>Cyber Crimes and Enforcement:</b> Cyber Appellate Tribunal, Cyber Crimes-Cyber Contraventions, Cyber Offences, Power of Investigation & Search, E-Evidence and Computer Forensic <b>Emerging Issues and Legal Considerations:</b> ISP & Intermediary Not to be Liable in Certain Cases, Consequential Amendments in Various Conventional Laws in India, Grey Areas of Information Technology Act, 2000, <b>Jurisdiction and Privacy:</b> Cyber Jurisdiction, E-Consumers, Privacy of Online Data and Information	<b>15</b>
<b>III</b>	<b>Intellectual Property Rights and Online Regulations:</b> Free Speech Online or Online Freedom of Speech and Expression and Liability of Intermediary <b>Intellectual Property Rights (IPRs), Copyrights &amp; Patents:</b> International and Indian Scenario, Copyright Issues and Digital Medium, Patent Issues in Digital Medium <b>Disputes and Resolution:</b> Domain Name Dispute & Resolution and Trademark Issues in Digital Medium, Spamming and Phishing	<b>15</b>
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Cyber Laws &amp; Information Technology (For LL.B.) Paperback – 1 January 2020</li> <li>2. Cyber Law in India, Satish Chandra, ABS Books, 2017</li> <li>3. Cyber Security and Cyber Laws, Nilakshi Jain, Wiley India, October 2020</li> </ol> <p><b>Additional Reference(s):</b></p> <ol style="list-style-type: none"> <li>1. Cyber Laws, Justice Yatindra Singh, Universal Law Publishing, Universal Publishing, 2016</li> <li>2. Cyber laws, Dr. Gupta &amp; Agrawal, PREMIER PUBLISHING COMPANY, 2022</li> <li>3. Cyber Law - An Exhaustive Section Wise Commentary On The Information Technology, Pavan Duggal, Universal Publishing (LexisNexis), 2nd Edition, 2017</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP605</b>	<b>Project Work – II</b>	<b>2</b>	<b>3</b>
<b><u>Refer to the Project Guidelines mentioned at the end</u></b>			

## Evaluation Scheme

### I. Internal Evaluation for Theory Courses – 25 Marks

#### (i) Mid-Term Class Test – 15 Marks

- It should be conducted using any **learning management system** such as **Moodle** (Modular object-oriented dynamic learning environment)
- The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes**.

#### (ii) Assignment/ Case study/ Presentations – 10 Marks

- Assignment / Case Study Report / Presentation can be uploaded on any **learning management system**.

### II. External Examination for Theory Courses – 75 Marks

- Duration: **2.5 Hours**
- Theory question paper pattern:

<b>All questions are compulsory.</b>			
<b>Question</b>	<b>Based on</b>	<b>Options</b>	<b>Marks</b>
Q.1	Unit I	<i>Any 4 out of 6</i>	20
Q.2	Unit II	<i>Any 4 out of 6</i>	20
Q.3	Unit III	<i>Any 4 out of 6</i>	20
Q.4	Unit I, II and III	<i>Any 5 out of 6</i>	15

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be sub-divided into sub questions as a, b, c, d, etc. & the allocation of Marks depends on the weightage of the topic.

### III. Practical Examination

- Each subject carries 50 Marks  
**40 marks + 05 marks (journal) + 05 marks (viva)**
- Duration: **2 Hours** for each practical course.
- Minimum **80% practical** from each core subjects are required to be completed.
- **Certified Journal is compulsory for appearing at the time of Practical Exam**
- The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

#### **IV. Project**

The evaluation of the project will include a viva voce, which will assess the project based on the following parameters:

- **Documentation – 30 Marks:** The completeness, accuracy, and professionalism of the project documentation, including the project report and supporting materials, will be considered.
- **Quality of the Project – 15 Marks:** The overall quality of the project, including its design, implementation, and user experience, will be evaluated.
- **Working of the Project – 20 Marks:** The functionality and performance of the project will be assessed to determine how well it meets the specified requirements and objectives.
- **Project Presentation – 15 Marks:** The clarity, organization, and effectiveness of the project presentation will be evaluated.
- **Viva – 20 Marks:** The viva voce session will provide an opportunity for the student to demonstrate their knowledge and understanding of the project, as well as to answer questions and engage in a discussion with the evaluators.

## **Project Guidelines (for USCSP505 and USCSP605)**

### **Aim:**

The Project Work as part of B.Sc. Computer Science program provides students with practical experience in applying their knowledge and skills to real-world projects, emphasizing hands-on experience in industry-standard project practices. It focuses on project development, implementation, and deployment using computer science principles and techniques. Students will work individually or in teams to design, develop, and present a substantial software project, gaining exposure to real-life project scenarios. It also covers project planning, requirements gathering, software design, coding, testing, debugging, documentation, and project management, following industry best practices. Through these projects, students will enhance their problem-solving abilities, gain proficiency in software development methodologies, and strengthen their practical skills in computer science.

### **Objectives:**

- Apply interdisciplinary knowledge to effectively solve real-life problems using acquired skills and concepts.
- Gain hands-on experience in the software development life cycle, encompassing requirements analysis, design, implementation, testing, and deployment.
- Familiarize with global IT industry standards, ethics, and professional practices to thrive in a professional environment.
- Develop teamwork and project management skills through structured collaboration, effective communication, and task delegation.
- Produce professional technical documentation aligning with industry practices, ensuring clarity, accuracy, and usability.
- Acquire time management, resource allocation, and personnel coordination skills for efficient project execution.

### **Project Types:**

- a) Developing a solution for a real-life problem:** In this case, the project focuses on addressing an existing requirement for a computer-based solution that has practical applications. The project should successfully implement the different stages of the system development life cycle. Examples: Secure Online Banking System, Machine Learning-based Disease Diagnosis System, Cloud-based Document Management System.
- b) Innovative Product Development:** These projects involve exploring and developing a computer-based solution with a unique and innovative utility. Examples: Cybersecurity Monitoring and Threat Detection System, Machine Learning-powered Predictive Maintenance System for Industrial Equipment, IoT-based Smart Energy Management System.
- c) Research-Level Project:** These projects involve conducting research and development to explore advanced technologies and solve complex problems. Examples: Deep Learning-based Image Recognition System for Medical Imaging, Cloud Computing Infrastructure Optimization for Big Data Processing, Data Science-driven Predictive Analytics for Sales Forecasting. The methodology and reporting of such projects may vary based on the project supervisor's guidance.

## **Tools & Technologies:**

In the project work, students are granted complete freedom to select platforms, tools, and programming languages without any imposed restrictions. This approach encourages creativity, flexibility, and exploration of various technologies. By prioritizing open-source technologies, students can leverage a vast array of resources and community support. Commonly employed tools include IDEs, version control systems (e.g., Git), programming languages (e.g., Python, Java), databases (e.g., MySQL), and web frameworks (e.g., Django, Ruby on Rails). The evaluation process focuses on the project's content and implementation rather than the specific tools chosen, ensuring a fair assessment of the students' skills and problem-solving abilities.

## **Project Guide:**

Assigning a project guide to each project or group is a mandatory requirement to ensure the successful completion of the project work. The guide plays a crucial role as a mentor and technical expert, providing invaluable support and guidance to students. They are expected to facilitate effective communication and teamwork, review project proposals, assign schedules, and monitor progress on a regular basis. Additionally, guides are expected to offer timely feedback, provide guidance on project planning and implementation strategies, evaluate the quality of work, and promote professionalism and ethical conduct. Their expertise and involvement are essential in helping students navigate challenges, make informed decisions, and achieve their project goals effectively.

## **Project Team Size:** 1 – 2 members

**Project Proposal:** The project proposal is a mandatory document that serves as a foundation for the project. It helps students define their project idea, receive early evaluation and feedback, establish clear communication with the project guide, and take ownership of the project's successful execution. A formal proposal ensures systematic and professional project planning, fostering critical thinking, effective communication, and project management skills. The proposal provides a roadmap and increases the chances of a successful outcome. Before initiating a project, it is mandatory to submit a project proposal for approval. **The original duly approved project proposal should be attached to the final project report.** The project proposal for UG computer science projects should include the following contents:

- Title
- Introduction
- Objectives: Clearly state the objectives of the project. What specific goals do you aim to achieve?
- Scope
- Methodology
- Tools and Technologies
- Timeline
- Resources
- Expected Outcomes
- References



## **Project Report:**

The Certified Copy of Hard Bound Project Report must adhere to the following guidelines:

- No of Copies: Team Size + 1 (College / Department Copy)
- The project report should include the following
  - Title Page (*Sample attached in Appendix*)
  - Certificate (*Sample attached in Appendix*)
  - Declaration (*Sample attached in Appendix*)
  - Acknowledgement
  - Table of Contents
  - Original Copy of approved Project Proposal
  - Self-attested copy of Plagiarism Report from any open source tool.
  - Chapters / Sections depending upon the type of project
  - List of Tables and/or List of Figures
  - References (IEEE / Springer format)
  - Glossary
  - Appendices (Survey datasheets / Questionnaires, ect)
- Use of LaTeX for documentation purposes should be preferred.
- The text of the report should be set in 12 pt, Times New Roman font, and single-spaced.
- Chapter headings should be centered, written in 20 pt, Times New Roman font, bold, and in all caps.
- These guidelines ensure a standardized format for the project report, promoting clarity and readability.

**SAMPLE TITLE PAGE FORMAT**

A PROJECT REPORT

on

**<PROJECT NAME>**

*Submitted by*

**Mr. XYZ**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF SCIENCE**

in

**COMPUTER SCIENCE**

*under the guidance of*

**<Guide Name>**

**Department of Computer Science**

<<College Logo>>

<<College Name>>

**(Sem V / VI)**

**(202- – 202-)**

**SAMPLE CERTIFICATE FORMAT**

<<College Logo>>

<<**College Name**>>>,

<<College Address>>

**Department of Computer Science**

**CERTIFICATE**

This is to certify that Mr./Ms. \_\_\_\_\_ of **T.Y.B.Sc. (Sem V/VI)** class has satisfactorily completed the Project \_\_\_\_\_, to be submitted in the partial fulfillment for the award of **Bachelor of Science in Computer Science** during the academic year **202- – 202-**.

**Date of Submission:**

**Project Guide**

**Head / Incharge,  
Department Computer Science**

**College Seal**

**Signature of Examiner**

**SAMPLE DECLARATION FORMAT**

**DECLARATION**

I, \_\_\_\_\_, hereby declare that the project entitled  
“ \_\_\_\_\_ ” submitted in the partial  
fulfillment for the award of **Bachelor of Science in Computer Science** during the academic year  
**202- – 202-** is my original work and the project has not formed the basis for the award of any  
degree, associateship, fellowship or any other similar titles.

**Signature of the Student:**

**Place:**

**Date:**