

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 15th 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 23rd June, 2016 has been accepted by the Academic Council at its meeting held on, 24th 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) 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
27th 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 27th 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

Course: USCS101	Computer Organization and Design (Credits : 2 Lectures/Week: 3)	
<p>Objectives. To understand the structure and operation of modern processors and their instruction sets</p> <p>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) To learn about how computer systems work and underlying principles 2) To understand the basics of digital electronics needed for computers 3) To understand the basics of instruction set architecture for reduced and complex instruction sets 4) To understand the basics of processor structure and operation 5) To understand how data is transferred between the processor and I/O devices 		
Unit I	<p>Computer Abstractions and Technology: Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters.</p> <p>Logic circuits and functions: 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>Instruction set architectures: 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>Basic Processor Unit: 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>Basic I/O: 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"> 1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011 2. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010 	
--	--

Course: USCS102	Programming with Python- I (Credits : 2 Lectures/Week: 3)	
<p>Objectives The objective of this paper is to introduce various concepts of programming to the students using Python.</p> <p>Expected learning outcomes</p> <ol style="list-style-type: none"> 1) Students should be able to understand the concepts of programming before actually starting to write programs. 2) Students should be able to develop logic for Problem Solving. 3) Students should be made familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc. 4) Students should be able to apply the problem solving skills using syntactically simple language i.e. Python (version: 3.X or higher) 		
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

Course: USCS103	Free and Open-source Software (Credits : 2 Lectures/Week: 3)	
<p>Objective: 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>Expected Learning Outcome:</p> <ol style="list-style-type: none"> 1) Upon completion of this course, students should have a good working knowledge of Open Source ecosystem, its use, impact and importance. 2) This course shall help student to learn Open Source methodologies, case studies with real life examples. 		
Unit I	<p>Introduction 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>Methodologies 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>	15L

	<p>Social Impact 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>Case Studies 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>Collaboration, Community and Communication</p> <p>Contributing to Open Source Projects 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>Understanding Open Source Ecosystem 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>Text books:</p> <ol style="list-style-type: none"> 1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006 2. The official Ubuntu Book, 8th Edition <p>Additional references:</p> <ol style="list-style-type: none"> 1. The Linux Documentation Project: http://www.tldp.org/ 2. Docker Project Home: http://www.docker.com 3. Linux kernel Home: http://kernel.org 4. Open Source Initiative: https://opensource.org/ 5. Linux Documentation Project: http://www.tldp.org/ 6. Wikipedia: https://en.wikipedia.org/ 7. https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia 8. Github: https://help.github.com/ 9. The Linux Foundation: http://www.linuxfoundation.org/ 		

Course: USCS104	Database Systems (Credits : 2 Lectures/Week: 3)	
<p>Objectives: 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>Expected Learning Outcomes</p> <ol style="list-style-type: none"> 1) Students should be able to evaluate business information problem and find the requirements of a problem in terms of data. 2) Students should be able to design the database schema with the use of appropriate data types for storage of data in database. 3) Students should be able to create, manipulate, query and back up the databases. 		
Unit I	<p>Introduction to DBMS – Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture</p> <p>Data models - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network)</p> <p>Entity Relationship Model - 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>Relational data model– 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>ER to Table- Entity to Table, Relationship to tables with and without key constraints.</p>	15L
Unit II	<p>Schema refinement and Normal forms: Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition.</p> <p>Relational Algebra operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division)</p> <p>DDL Statements - 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>DML Statements – 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>Functions – 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>Joining Tables – inner join, outer join (left outer, right outer, full outer)</p> <p>Subqueries – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries</p> <p>Database Protection: Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control</p> <p>Views (creating, altering dropping, renaming and manipulating views)</p> <p>DCL Statements (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges)</p>	15L
<p>Text books:</p> <ol style="list-style-type: none"> 1. Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010 2. Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill, 2007 3. Joel Murach, Murach’s MySQL, Murach, 2012 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press, 2005. 		

Course: USCS105	Discrete Mathematics (Credits : 2 Lectures/Week: 3)	
<p>Objectives: 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>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) To provide overview of theory of discrete objects, starting with relations and partially ordered sets. 2) Study about recurrence relations, generating function and operations on them. 3) Give an understanding of graphs and trees, which are widely used in software. 4) Provide basic knowledge about models of automata theory and the corresponding formal languages. 		

Unit I	<p>Recurrence Relations</p> <p>(a) Functions: 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) Relations: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Daigrams , Maximum and Minimum elements, Lattices</p> <p>(c) Recurrence Relations: 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>Counting Principles , Languages and Finite State Machine</p> <p>(a) Permutations and Combinations: 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) Counting Principles: 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>(c) Languages, Grammars and Machines: Languages , regular Expression and Regular languages, Finite state Automata, grammars, Finite state machines, Gödel numbers, Turing machines.</p>	15L
Unit III	<p>Graphs and Trees</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, Traversing a graph- Breadth-First search and Depth-First search.</p> <p>(b) Trees: 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.

Course: USCS106	Descriptive Statistics and Introduction to Probability (Credits : 2 Lectures/Week: 3)	
Objectives: 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.		
Expected Learning Outcomes: 1) Enable learners to know descriptive statistical concepts 2) Enable study of probability concept required for Computer learners		
Unit I	Data Presentation Data types : attribute, variable, discrete and continuous variable Data presentation : frequency distribution, histogram o give, curves, stem and leaf display Data Aggregation 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	Moments: raw moments, central moments, relation between raw and central moments Measures of Skewness and Kurtosis: based on moments, quartiles, relation between mean, median, mode for symmetric, asymmetric frequency curve. Correlation and Regression: bivariate data, scatter plot, correlation, nonsense correlation, Karl pearson's coefficients of correlation, independence. Linear regression: fitting of linear regression using least square regression, coefficient of determination, properties of regression coefficients (only statement)	15L

Unit III	<p>Probability : Random experiment, sample space, events types and operations of events</p> <p>Probability definition : classical, axiomatic, Elementary Theorems of probability (without proof)</p> <ul style="list-style-type: none"> – $0 \leq P(A) \leq 1$, – $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ – $P(A') = 1 - P(A)$ – $P(A) \leq P(B)$ if $A \subset B$ <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. 6th Edⁿ 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.

Course: USCS107	Soft Skills Development (Credits : 2 Lectures/Week: 3)	
<p>Objectives:</p> <p>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>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) To know about various aspects of soft skills and learn ways to develop personality 2) Understand the importance and type of communication in personal and professional environment. 3) To provide insight into much needed technical and non-technical qualities in career planning. 4) Learn about Leadership, team building, decision making and stress management 		

Unit I	<p>Introduction to Soft Skills and Hard Skills</p> <p>Personality Development: Knowing Yourself, Positive Thinking, Johari's Window, Communication Skills, Non-verbal Communication, Physical Fitness</p> <p>Emotional Intelligence: 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>Etiquette and Mannerism: Introduction, Professional Etiquette, Technology Etiquette</p> <p>Communication Today: 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>Academic Skills</p> <p>Employment Communication: Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter</p> <p>Professional Presentation: Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation</p> <p>Job Interviews: 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>Group Discussion: 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>Professional Skills</p> <p>Creativity at Workplace: Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method</p> <p>Ethical Values: 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>Capacity Building: Learn, Unlearn and Relearn: Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building</p> <p>Leadership and Team Building: Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams,</p> <p>Decision Making and Negotiation: Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts</p> <p>Stress and Time Management: 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>Computer Organization and Design</p> <ol style="list-style-type: none"> 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. 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. 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." <p># Practical No. 1 to 8 can be performed using any open source simulator (like Logisim) (Download it from https://sourceforge.net/projects/circuit/)</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 https://sourceforge.net/projects/spimsimulator/</p>	

USCSP102	<p>Programming with Python – I</p> <ol style="list-style-type: none"> 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. 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. 3. Programs based on lists, conditional constructs, the for statement and the range function; interactively using the built-in functions len, sum, max, min 4. Programs related to string manipulation 5. Programs based on the while statement; importing and executing built-in functions from the time, math and random modules 6. Programs using break and continue statements. 7. Programs related to dictionaries 8. Programs using list comprehensions and anonymous functions 9. Programs using the built-in methods of the string, list and dictionary classes 	
-----------------	---	--

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

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

	<ol style="list-style-type: none"> 5. Perform the following: <ul style="list-style-type: none"> • Simple Queries • Simple Queries with Aggregate functions • Queries with Aggregate functions (group by and having clause) 6. Queries involving <ul style="list-style-type: none"> • Date Functions • String Functions • Math Functions 7. Join Queries <ul style="list-style-type: none"> • Inner Join • Outer Join 8. Subqueries <ul style="list-style-type: none"> • With IN clause • With EXISTS clause 9. Views <ul style="list-style-type: none"> • Creating Views (with and without check option) • Dropping views • Selecting from a view 10. DCL statements <ul style="list-style-type: none"> • Granting and revoking permissions 	
<p style="text-align: center;">USCSP105</p>	<p>Discrete Mathematics</p> <ol style="list-style-type: none"> 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. 2. Partial ordering sets, Hasse diagram and Lattices. 3. Recurrence relation. 4. Different counting principles. 5. Finite state Automata and Finite state machines. 6. Warshall's Algorithm. 7. Shortest Path algorithms. 8. Operations on graph. 9. Breadth and Depth First search algorithms. 10. Concept of searching, inserting and deleting from binary search trees. 	
<p style="text-align: center;">USCSP106</p>	<p>Descriptive Statistics and Introduction to Probability <i>(To be implemented using R)</i></p> <ol style="list-style-type: none"> 1. Frequency distribution and data presentation 2. Measures of central tendency 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. 4. Frequency distribution using cut(), table() 5. Data presentation 6. Summary Statistics (measures of central tendency, dispersion) 7. Measures of skewness and kurtosis 8. Correlation and regression 9. Probability 10. Conditional probability 	

Semester II - Theory

Course: USCS201	Programming with C (Credits : 2 Lectures/Week: 3)	
<p>Objectives: 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>Expected Learning Outcomes</p> <ol style="list-style-type: none"> 1) Students should be able to write, compile and debug programs in C language. 2) Students should be able to use different data types in a computer program. 3) Students should be able to design programs involving decision structures, loops and functions. 4) Students should be able to explain the difference between call by value and call by reference 5) Students should be able to understand the dynamics of memory by the use of pointers. 6) Students should be able to use different data structures and create/update basic data files. 		
Unit I	<p>Structure of C program: Header and body, Use of comments. Interpreters vs compilers, Python vs C. Compilation of a program. Formatted I/O: printf(), scanf().</p> <p>Data: 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>Variables: 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>Types of operators: 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>Iterations: 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>Arrays: (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>Data Input and Output functions: Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().</p> <p>Manipulating Strings: Declaring and initializing String variables, Character and string handling functions. Compare with Python strings.</p> <p>Functions: Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values.</p> <p>Recursion: Definition, Recursive functions.</p>	15L

Unit III	<p>Pointer: 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>Dynamic Memory Allocation: malloc(), calloc(), realloc(), free() and sizeof operator. Compare with automatic garbage collection in Python.</p> <p>Structure: 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>Unions: Defining and working with unions.</p> <p>File handling: 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>Text books:</p> <ol style="list-style-type: none"> 1. Programming in ANSI C (Third Edition) : E Balagurusamy, TMH <p>Additional References:</p> <ol style="list-style-type: none"> 1. Pradip Dey, Manas Ghosh, "Programming in C", second edition, Oxford University Press 2. Yashavant P. Kanetkar. " Let Us C", BPB Publications 		

Course: USCS202	Programming with Python – II (Credits : 2 Lectures/Week: 3)	
<p>Objective: 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>Expected Learning Outcomes</p> <ol style="list-style-type: none"> 1) Students should be able to understand how to read/write to files using python. 2) Students should be able to catch their own errors that happen during execution of programs. 3) Students should get an introduction to the concept of pattern matching. 4) Students should be made familiar with the concepts of GUI controls and designing GUI applications. 5) Students should be able to connect to the database to move the data to/from the application. 6) 6)Students should know how to connect to computers, read from URL and send email. 		
Unit I	<p>Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problemsolving applications.</p> <p>Exception handling: What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise.</p> <p>Regular Expressions: Concept of regular expression, various types of regular expressions, using match function.</p>	15 L
Unit II	<p>GUI Programming in Python (using Tkinter/wxPython/Qt) 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>Database connectivity in Python: Installing mysql connector, accessing connector module module, using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity.</p> <p>Network connectivity: Socket module, creating server-client programs, sending email, reading from URL</p>	15 L
<p>Text books:</p> <ol style="list-style-type: none"> 1. Paul Gries , Jennifer Campbell, Jason Montojo, <i>Practical Programming: An Introduction to Computer Science Using Python 3</i>, Pragmatic Bookshelf, 2/E 2014 <p>Additional References:</p> <ol style="list-style-type: none"> 1. James Payne , <i>Beginning Python: Using Python 2.6 and Python 3</i>, Wiley India, 2010 2. A. Lukaszewski, <i>MySQL for Python: Database Access Made Easy</i>, Pact Publisher, 2010 		

Course: USCS203	Linux (Credits : 2 Lectures/Week: 3)	
<p>Objectives: 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>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 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. 2) This course shall help student to learn advanced subjects in computer science practically. 3) Student shall be able to progress as a Developer or Linux System Administrator using the acquired skill set. 		
Unit I	<p>Introduction 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>Installation Installation methods, Hands on Installation using CD/DVD or USB drive.</p> <p>Linux Structure 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>Graphical Desktop Session Management, Basic Desktop Operations, Network Management, Installing and Updating Software, Text editors: gedit, vi, vim, emacs, Graphics editors, Multimedia applications.</p> <p>Command Line Command line mode options, Shells, Basic Commands, General Purpose Utilities, Installing Software, User management, Environment variables, Command aliases.</p> <p>Linux Documentation man pages, GNU info, help command, More documentation sources</p> <p>File Operations Filesystem, Filesystem architecture, File types, File attributes, Working with files, Backup, compression</p>	15L
Unit III	<p>Security Understanding Linux Security, Uses of root, sudo command, working with passwords, Bypassing user authentication, Understanding ssh</p> <p>Networking 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>Basic Shell Scripting 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/>

Course: USCS204	Data Structures (Credits : 2 Lectures/Week: 3)	
<p>Objectives: 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>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) Learn about Data structures, its types and significance in computing 2) Explore about Abstract Data types and its implementation 3) Ability to program various applications using different data structure in Python 		
Unit I	<p>Abstract Data Types: Introduction, The Date Abstract Data Type, Bags, Iterators. Application</p> <p>Arrays: Array Structure, Python List, Two Dimensional Arrays, Matrix Abstract Data Type, Application</p> <p>Sets and Maps: 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>Algorithm Analysis: Complexity Analysis-Big-O Notation, Evaluating Python Code, Evaluating Python List, Amortized Cost, Evaluating Set ADT, Application</p> <p>Searching and Sorting: 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>Linked Structures: 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>Stacks: Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-Balanced Delimiters, Evaluating Postfix Expressions</p> <p>Queues: Queue ADT, Implementing Queue-Using Python List, Circular Array, Using List, Priority Queues- Priority Queue ADT, Bounded and unbounded Priority Queues</p> <p>Advanced Linked List: Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation, Multi Lists</p>	15L

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

Course: USCS205	Calculus (Credits : 2 Lectures/Week: 3)	
<p>Objectives: 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>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) Understanding of Mathematical concepts like limit, continuity, derivative, integration of functions. 2) Ability to appreciate real world applications which uses these concepts. 3) Skill to formulate a problem through Mathematical modeling and simulation. 		
Unit I	<p>DERIVATIVES AND ITS APPLICATIONS: 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>INTEGRATION AND ITS APPLICATIONS: 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>PARTIAL DERIVATIVES AND ITS APPLICATIONS: 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.

Course: USCS206	Statistical Methods and Testing of Hypothesis (Credits : 2 Lectures/Week: 3)	
Objectives: 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.		
Expected Learning Outcomes: 1) Enable learners to know descriptive statistical concepts 2) Enable study of probability concept required for Computer learners		
Unit I	Standard distributions: 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	Hypothesis testing: 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	Non-parametric tests: need of non-parametric tests, sign test, Wilcoxon's signed rank test, run test, Kruskal-Walis tests. Post-hoc analysis of one-way analysis of variance : Duncan's test Chi-square test of association	15L
Text Book: 1. Trivedi, K.S.(2009) : 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 th Ed ⁿ 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 th Edition) : Applied Statistics, S. Chand and Son's, New Delhi 5. Montgomery, D.C. (2001): Planning and Analysis of Experiments, Wiley.		

Course: USCS207	Green Technologies (Credits : 2 Lectures/Week: 3)	
Objectives: 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>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) Learn about green IT can be achieved in and by hardware, software, network communication and data center operations. 2) Understand the strategies, frameworks, processes and management of green IT 		
Unit I	<p>Green IT Overview: 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>Green Devices and Hardware: Introduction , Life Cycle of a Device or Hardware, Reuse, Recycle and Dispose</p> <p>Green Software: Introduction , Processor Power States , Energy-Saving Software Techniques, Evaluating and Measuring Software Impact to Platform Power</p> <p>Sustainable Software Development: Introduction, Current Practices, Sustainable Software, Software Sustainability Attributes, Software Sustainability Metrics, Sustainable Software Methodology, Defining Actions</p>	15L
Unit II	<p>Green Data Centres: 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>Green Data Storage: Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management</p> <p>Green Networks and Communications: Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards</p> <p>Enterprise Green IT Strategy: 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>Sustainable Information Systems and Green Metrics: 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>Enterprise Green IT Readiness: Introduction, Readiness and Capability, Development of the G-Readiness Framework, Measuring an Organization's G-Readiness</p> <p>Sustainable IT Services: Creating a Framework for Service Innovation: Introduction, Factors Driving the Development of Sustainable IT, Sustainable IT Services (SITS), SITS Strategic Framework</p> <p>Green Enterprises and the Role of IT: 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>Text book:</p> <ol style="list-style-type: none"> 1) <i>Harnessing Green IT: Principles and Practices</i>, San Murugesan, G. R. Ganadharan, Wiley & IEEE. <p>Additional References:</p> <ol style="list-style-type: none"> 1) <i>Green IT</i>, Deepak Shikarpur, Vishwkarma Publications, 2014 2) <i>Green Communications: Principles, Concepts and Practice</i>- Samdanis et al, J. Wiley 3) <i>Green IT for Sustainable Business Practice: An ISEB Foundation Guide</i>, Mark G. O'Neill, The Chartered Institute for IT, 2010 		

Semester II – Practical

Course: USCSP2	Practical of USCS201 + USCS202 + USCS203+USCS204+ USCS205+ USCS206 (Credits : 6, Lectures/Week: 18)	
USCSP201	Programming with C <ol style="list-style-type: none">1. Programs to understand the basic data types and I/O.2. Programs on Operators and Expressions3. Programs on decision statements.4. Programs on looping.5. Programs on arrays.6. Programs on functions.7. Programs on structures and unions.8. Programs on pointers.9. Programs on string manipulations.10. Programs on basic file operations.	
USCSP202	Programming with Python-II <ol style="list-style-type: none">1. Programs to read and write files.2. Programs with iterables and iterators.3. Program to demonstrate exception handling.4. Program to demonstrate the use of regular expressions.5. Program to show draw shapes & GUI controls.6. Program to create server-client and exchange basic information.7. Program to send email & read contents of URL.	

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

	<ol style="list-style-type: none"> 7. Use environment <ol style="list-style-type: none"> a. Which account are you logged in? How do you find out? 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. c. Get you current working directory. d. Explore different ways of getting command history, how to run previously executed command without typing it? e. Create alias to most commonly used commands like. 8. Linux Editors: <code>vim/emacs</code> <ol style="list-style-type: none"> a. Create,modify, search, navigate a file in editor. b. Learn all essential commands like search, search/replace, highlight, show line numbers. 9. Linux Security: <ol style="list-style-type: none"> a. Use of <code>sudo</code> to change user privileges to root b. Identify all operations that require <code>sudo</code> privileges c. Create a new user and add it to <code>sudo</code> configuration file. d. Set password for new user. e. Modify the expiration date for new user using password ageing. f. Delete newly added user. 10. Network: <ol style="list-style-type: none"> a. Get IP address of your machine using <code>ifconfig</code>. b. If IP is not set, then assign an IP address according to your network settings. c. Get hostname of your machine. d. Use <code>ping</code> to check the network connectivity to remote machines. e. Use <code>telnet/ssh</code> to connect to remote machines and learn the difference between the two. f. Troubleshooting network using <code>traceroute</code>, <code>ping</code>, <code>route</code> commands. 11. Shell Scripting <ol style="list-style-type: none"> a. Searching with <code>grep</code>: Search for your username in the <code>/etc/passwd</code> file. 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? 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>. 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> 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. 	
--	---	--

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

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



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

Semester – III				
Course Code	Course Type	Course Title	Credits	Lectures/Week
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

Semester – IV				
Course Code	Course Type	Course Title	Credits	Lectures/Week
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To learn basic concepts and structure of operating systems • To learn about process and synchronization in operating system level • To learn CPU scheduling algorithms • To learn Memory and File system management 			
<p>Learning Outcomes:</p> <p>After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Work with any type of operating system • Handle threads, processes, process synchronization • Implement CPU scheduling algorithms • Understand the background role of memory management • Design file system. 			
Unit	Topics	No of Lectures	
I	<p>Introduction to Operating-Systems: Definition of Operating System, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments</p> <p>Operating-System Structures: Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure</p> <p>Processes: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication</p> <p>Threads: Overview, Multicore Programming, Multithreading Models</p>	15	
II	<p>Process Synchronization: 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>CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling</p> <p>Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock</p>	
III	<p>Main Memory: Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table</p> <p>Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing</p> <p>Mass-Storage Structure: Overview, Disk Structure, Disk Scheduling, Disk Management</p> <p>File-System Interface: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing</p> <p>File-System Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management</p>	15
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 2021 <p>Additional Reference(s):</p> <ol style="list-style-type: none"> 1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill, 2017 2. Naresh Chauhan, Principles of Operating Systems, Oxford Press, 2014 3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016 		

Course Code	Course Title	Credits	Lectures /Week
USCSP301	Principles of Operating Systems – Practical	1	3
1	Process Communication: <ol style="list-style-type: none"> Write a program to give a solution to the producer–consumer problem using shared memory. Write a program to give a solution to the producer–consumer problem using message passing. 		
2	Threads: <ol style="list-style-type: none"> Write a program to work with a single thread. Write a program to work with multi threads. 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. 		
3	Synchronization: <ol style="list-style-type: none"> Write a program to give a solution to the Bounded buffer problem. Write a program to give a solution to the readers–writers problem. 		
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To offer the learner the relevant Linear Algebra concepts through Computer Science applications. • To interpret existence and analyze the solution set of a system of linear equations. • To formulate, solve, apply, and interpret properties of linear systems. • To learn about the concept of linear independence of vectors over a field, and the dimension of a vector space. • To interpret basic concepts of linear transformations, dimension, matrix representation of a linear transformation, and the change of coordinate matrix. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Appreciate the relevance and applications of Linear Algebra in the field of Computer Science. • Understand the concepts through program implementation. • Instill a computational thinking while learning linear algebra. • Express clear understanding of the concept of a solution to a system of equations. • Find eigenvalues and corresponding eigenvectors for a square matrix. 			
Unit	Topics	No of Lectures	
I	<p>Field: Introduction to complex numbers, complex numbers in Python, abstracting over fields, Playing with GF (2).</p> <p>Vectors: 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>The Vector Space: Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise</p>	15	
II	<p>Matrix: 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>Basis: 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>Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator</p> <p>Gaussian elimination: Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination.</p>	
III	<p>Inner Product: The inner product for vectors over the reals, Orthogonality.</p> <p>Orthogonalization: Projection orthogonal to multiple vectors, projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, orthogonal complement.</p> <p>Eigenvalues and Eigenvectors: 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>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Coding the Matrix Linear Algebra through Applications to Computer Science, First Edition, Philip N. Klein, Newtonian Press 2013 2. Schaum's Outline of Linear Algebra, Sixth Edition by Seymour Lipschutz, Marc Lipson, McGraw Hill 2017 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Linear Algebra and Probability for Computer Science Applications, First Edition, Ernest Davis, A K Peters/CRC Press, 2012. 2. Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition, 2007 3. Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition, 2002 4. Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2008. 5. Computer Networking With Internet Protocols and Technology, William Stallings, Pearson Education India, 2013. 		

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"> Addition of two complex numbers Displaying the conjugate of a complex number Plotting a set of complex numbers Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number $a = 1/2$, $a = 1/3$, $a = 2$ etc. 		
2	Write a program to do the following: <ul style="list-style-type: none"> Enter a vector u as a n-list Enter another vector v as a n-list Find the vector $au + bv$ for different values of a and b Find the dot product of u and v 		
3	Vector Applications: Classify given data using support vector machines (SVM)		
4	Basic Matrix Operations: <ul style="list-style-type: none"> Matrix Addition, Subtraction, Multiplication Check if matrix is invertible. If yes then find Inverse 		
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"> 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 		
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To introduce data abstraction and data representation in memory • To describe, design and use of elementary data structures such as stack, queue, linked list, tree and graph • How and why different data structures are used for different types of problems. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to-</p> <ul style="list-style-type: none"> • Create different types of data structures. • Understand which data structure to be used based on the type of the problem. • Apply combined knowledge of algorithms and data structures to write highly effective programs in various domains. 			
Unit	Topics	No of Lectures	
I	<p>Abstract Data Type: Different Data Types, different types of data structures & their classifications, Introduction to ADT, Creating user-specific ADT</p> <p>Linked Structures: ADT for linked list, Advantages & Disadvantages, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, applications of linked list like polynomial equation</p> <p>Stacks: Stack ADT for Stack, Advantages & Disadvantages, Applications of stack like balanced delimiter, prefix to postfix notation</p> <p>Queues: Queue ADT, Advantages & Disadvantages, linked representations. Circular Queue operations, Dequeues, applications of queue like job scheduling queues</p>	15	
II	<p>Doubly Linked list: ADT of doubly linked list, Advantages & Disadvantages, Insertion and deletion of nodes at various positions</p> <p>Trees: ADT for Tree Structure. Advantages & disadvantages, Binary Tree-Properties, Implementation and Traversals, Binary Search Tree, Balanced BST, Threaded Binary Trees, AVL Trees, Applications of Tree like Huffman Coding,</p>		

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

Course Code	Course Title	Credits	Lectures /Week
USCSP303	Data Structures – Practical	1	3
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage. To develop understanding of Transaction management and crash recovery. To develop concepts of programming concepts of database. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> Master concepts of stored procedure, functions, cursors and triggers and its use. Learn about using PL/SQL for data management. Use efficiently Collections and records. Understand concepts and implementations of transaction management and crash recovery. 			
Unit	Topics	No of Lectures	
I	<p>Overview of PL/SQL: Advantages of PL/SQL, Main Features of PL/SQL, Architecture of PL/SQL</p> <p>Fundamentals of PL/SQL: 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>Control Statements: Conditional Selection Statements, LOOP Statements, Sequential Control Statements, GOTO, and NULL Statements.</p> <p>Sequences: creating sequences, referencing, altering, and dropping a sequence.</p> <p>Stored Procedures and Functions: 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>Collections and Records: 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>Error Handling: 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>Cursors: Overview of Cursor, Types of cursors, Invalid cursor Exception.</p> <p>Static and Dynamic SQL: 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>	
III	<p>Triggers: 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>Packages: Overview of a Package. Need of Packages, Package Specification, Package Body, Package Instantiation and Initialization.</p> <p>Transaction Management: 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>Crash Recovery: 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>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Mastering PL/SQL Through Illustrations: From Learning Fundamentals to Developing Efficient PL/SQL Blocks, Dr. B. Chandra, BPB Publication, 2020 2. Oracle Pl/Sql Training Guide., Training guide, BPB Publications, 2016 3. Raghu Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill,3rd Edition, 2014 4. Abraham Silberschatz, Henry F. Korth,S. Sudarshan , Database System Concepts, 6th Edition 2019 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Ivan Bayross, “SQL, PL/SQL -The Programming language of Oracle”, B.P.B. Publications 2009 2. Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, 2008 		

Course Code	Course Title	Credits	Lectures /Week
USCSP304	Advanced Database Concepts – Practical	1	3
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> To provide insight into java based applications using OOP concepts. To provide understanding of developing GUI based desktop applications in java. To provide knowledge of web based applications through servlet and jsp. To provide understanding and implementation of basic JSON 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> Design basic application in java using Graphical User Interface. The learner will be able to develop applications using swings The learner will be able to develop web based applications using servlet and jsp The learner will be able to connect databases with java through The learner will be able to perform programs using JSON objects 			
Unit	Topics	No of Lectures	
I	<p>Introduction: History, Features of Java, Java Development Kit, Java Application Programming Interface, Java Virtual Machine Java Program Structure, Java Tokens.</p> <p>OOPS: 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>Packages: Introduction to predefined packages, User Defined Packages, Access specifiers</p> <p>Exception Handling: Introduction, Pre-Defined Exceptions, try-catch-finally, throws, throw, User Defined Exceptions</p> <p>Multithreading: Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, wait() notify() notify all() methods</p>	15	
II	<p>Collection Framework: Introduction, java.util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes.</p>	15	

	<p>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, 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>Event Handling: Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes.</p> <p>JDBC: 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 & CLOB</p>	
<p style="text-align: center;">III</p>	<p>Servlets: 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>Java Server Pages (JSP): 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>JSON: Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java</p>	<p style="text-align: center;">15</p>
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Herbert Schildt, Java The Complete Reference, Eleventh Edition, McGraw-Hill Education, 2020 2. Bryan Basham, Kathy Sierra, Bert Bates, Head First Servlets and JSP, O’reilly (SPD), 2018 3. Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR, 2004 4. Ivan Bayross, Web Enabled Commercial Applications Development Using Java 2, BPB Publications 5. Java XML and JSON: Document Processing for Java SE by Jeff Friesen January 2019, Apress <p>Additional References:</p> <ol style="list-style-type: none"> 1. E. Balagurusamy, Programming with Java- A Primer, Tata McGraw-Hill Education India, 2014 2. Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press, 2018 3. Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) 4. Eric Jendrock, Jennifer Ball, D Carson and others, The Java EE 5 Tutorial, Pearson Education 5. The Java Tutorials: http://docs.oracle.com/javase/tutorial/ 6. Java Parsing Collection XML JSON: Map List XML JSON Transform by Yang Hu, 2019 		

Course Code	Course Title	Credits	Lectures /Week
USCSP305	Java based Application Development – Practical	1	3
1	<ul style="list-style-type: none"> a. Write a program to create a class and implement the concepts of Constructor Overloading, Method Overloading, Static methods b. Write a program to implement the concept of Inheritance and Method Overriding 		
2	<ul style="list-style-type: none"> a. Write a program to implement the concepts of Abstract classes and methods b. Write a program to implement the concept of interfaces 		
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"> a. List interface b. Set interface c. Map interface 		
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"> a. Write a JDBC program that displays the data of a given table b. Write a JDBC program to return the data of a specified record from a given table c. Write a JDBC program to insert / update / delete records into a given table 		
7	<ul style="list-style-type: none"> a. Construct a simple calculator using the JAVA Swings with minimum functionality. 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. 		
8	<ul style="list-style-type: none"> 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. b. Write a Servlet that displays the names and values of the cookie stored on the client. 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. 		
9	<ul style="list-style-type: none"> a. Write a registration Servlet that accepts the data for a given table and stores it in the database. b. Write a Servlet that displays all the records of a table. 		
10	<ul style="list-style-type: none"> 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. b. Write a JSP that displays the names and values of the cookie stored on the client. 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. 		

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

Course Code	Course Title	Credits	Lectures /Week
USCS306	Web Technologies	2	3
<p>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To understand the concepts of Hyper Text Markup Language and Cascading Style Sheets. • To learn JavaScript for creating dynamic websites. • To learn various operations performed on data among web applications using XML • To learn Server-Side Programming using PHP 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Design valid, well-formed, scalable, and meaningful pages using emerging technologies. • Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites • Develop and implement client-side and server-side scripting language programs. • Develop and implement Database Driven Websites. • Design and apply XML to create a markup language for data and document centric applications. 			
Unit	Topics	No of Lectures	
I	<p>HTML5: 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>CSS: 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>JavaScript: 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>XML: 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	
III	<p>AJAX: AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties and Methods, handling asynchronous requests using AJAX</p> <p>PHP: Variables and Operators, Program Flow, Arrays, working with Files and Directories, working with Databases, Working with Cookies, Sessions and Headers</p> <p>Introduction to jQuery: Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects.</p>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press, 2016 2. Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India, 2018 3. PHP: A Beginners Guide, Vikram Vaswani, TMH <p>Additional References:</p> <ol style="list-style-type: none"> 1. HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY, 2011 2. Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly, 2018 3. PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley, 2018 		

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"> a. Document Structure Tags b. Various Text Formatting Tags c. List Tags d. Image and Image Maps 		
2	Design a webpage that makes use of <ul style="list-style-type: none"> a. Table tags b. Form Tags (forms with various form elements) c. Navigation across multiple pages d. Embedded Multimedia elements 		
3	Design a webpage that make use of Cascading Style Sheets with <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> a. CSS b. XSL 		
7	Design a webpage to handle asynchronous requests using AJAX on <ul style="list-style-type: none"> a. Mouseover b. button click 		
8	Write PHP scripts for <ul style="list-style-type: none"> 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		

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

Course Code	Course Title	Credits	Lectures /Week
USCS3071	Creative Content Writing	2	3
<p>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To introduce students to the concepts of content writing. • To connect them with various writing and editing styles and techniques. • To help them develop their creative abilities. • To improve the learners' employability 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Understand the fundamentals of content creation for Blog, Website etc. • Acquire the ability to write and edit in a variety of styles and procedures • To develop the creative abilities. • To acquire essential language skills for editors. 			
Unit	Topics	No of Lectures	
I	<p>Basics of Content writing: Introduction to Content Writing, Learning Tone in Writing and Its Types, Comprehending style in writing and its Types, Common Grammatical Errors.</p> <p>Best Practices for Writing for the Web: 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>Things Marketers Write: The Ideal Length for Blog Posts, Podcast, Facebook Posts, Tweets, and Other Marketing Content.</p>	15	
II	<p>Social Media Writing: 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>Infographics: 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	

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

Course Code	Course Title	Credits	Lectures /Week
USCS3072	Green Technologies	2	3
<p>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • Know about Green IT Fundamentals: Business, IT, and the Environment • Green IT Strategies and Significance of Green IT Strategies • Green Enterprise Architecture and Green Information Systems • Sociocultural Aspects of Green IT and Green Compliance 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Explain drivers and dimensions of change for Green Technology • Appreciate Virtualization; smart meters and optimization in achieving green IT • Gain knowledge about green assets, green processes, and green enterprise architecture • ISO 14001 and related standards for Audit for Green Compliance 			
Unit	Topics	No of Lectures	
I	<p>Green IT Fundamentals: 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>Green IT Strategies: 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>Environmentally Responsible Business: 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>Green Assets: Introduction, Green Assets, Green IT Hardware, Green Data Centers and ICT Equipment, Server and Data Strategy</p>	15	
II	<p>Green Assets and emerging Trends: 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>Green Business Process Management: 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>Green Enterprise Architecture: 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;">III</p>	<p>Green Information Systems(GIS): Design and Development Models: Describing GIS, GIS Requirements</p> <p>Sociocultural Aspects of Green IT: 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>Green Compliance: Protocols, Standards, and Audits: 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>Emerging Carbon Issues: 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;">15</p>
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Green IT Strategies and Applications Using Environmental Intelligence, Bhuvan Unhelkar, CRC Press, 2016 2. Green Information and Communication Systems for a Sustainable Future, Rajshree Srivastava, Sandeep Kautish, Rajeev Tiwari. CRC Press, 2020 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Emerging Green Technologies, Matthew N. O. Sadiku, Taylor and Francis (CRC Press), 2022 2. Sustainability Awareness and Green Information Technologies, Tomayess Issa, Springer, 2021 3. Environmental Sustainability Role of Green Technologies, P. Thangavel, and G. Sridevi, Springer, 2016 		

Semester IV

Course Code	Course Title	Credits	Lectures /Week
USCS401	Theory of Computation	2	3
<p>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> To give an overview of the theoretical foundations of computer science from the perspective of formal languages To illustrate finite state machines to solve problems in computing To explain the hierarchy of problems arising in the computer sciences. To familiarize Regular grammars, context free grammar. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> Understand Grammar and Languages Learn about Automata theory and its application in Language Design Learn about Turing Machines and Pushdown Automata Understand Linear Bound Automata and its applications 			
Unit	Topics	No of Lectures	
I	<p>Automata Theory: 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>Formal Languages: 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>Regular Sets and Regular Grammar: Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar</p> <p>Context Free Languages: Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG</p> <p>Pushdown Automata: Definitions, Acceptance by PDA, PDA and CFG</p>	15	

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

Course Code	Course Title	Credits	Lectures /Week
USCSP401	Theory of Computation – Practical	1	3
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To Understand Basic Concepts of Networking. • To Understand Working of Network Layer Architecture. • To Learn Practical Implementation of Basic Routing Algorithms. • To Learn Different Networking Protocols. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Learn basic networking concepts and layered architecture. • Understand the concepts of networking, which are important for them to be known as a 'networking professionals'. 			
Unit	Topics	No of Lectures	
I	<p>Introduction: Networking standards and Administrations, networks, network types – LAN, MAN, WAN.</p> <p>Network Models: The OSI model, TCP/IP protocol suite,</p> <p>Introduction to Physical layer: Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance.</p> <p>Digital transmissions: Digital-to-digital conversion, analog-to-digital conversion, transmission modes</p> <p>Analog transmissions: digital-to-analog conversion, analog-to-analog conversion.</p> <p>Bandwidth Utilization – Multiplexing and Spectrum spreading: Multiplexing, Spread Spectrum</p> <p>Transmission media: Guided Media, Unguided Media</p> <p>Switching: Introduction, Circuit Switched Network, Packet Switching.</p>	15	
II	<p>Introduction to Data Link Layer: Link layer addressing, Data Link Layer Design Issues.</p> <p>Error detection and correction: -Block coding, cyclic codes, checksum, forward error correction, error correcting codes, error detecting codes.</p>	15	

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

Course Code	Course Title	Credits	Lectures /Week
USCSP402	Computer Networks – Practical	1	3
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To learn and understand the Concepts of Software Engineering • To learn and understand Software Development Life Cycle • To apply the project management and analysis principles to software project development. • To apply the design & testing principles to software project development. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • 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 • Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology. • Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice • Able to use modern engineering tools necessary for software project management, time management and software reuse. 			
Unit	Topics	No of Lectures	
I	<p>Introduction: 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>Requirement Analysis and System Modeling: 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;">II</p>	<p>System Design: 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>Software Measurement and Metrics: 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>Software Project Management: Estimation 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>Project Scheduling - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts</p>	<p style="text-align: center;">15</p>
<p style="text-align: center;">III</p>	<p>Risk Management - Risk strategies, Software risks, Risk Identification, projection, RMMM Quality Concepts</p> <p>Software Quality Assurance 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>Software Testing : 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;">15</p>
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Software Engineering, A Practitioner’s Approach, Roger S, Pressman, 2019 2. Software Engineering: principles and Practices, Deepak Jain, OXFORD University Press, 2008 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Software Engineering, Ian Sommerville, Pearson Education, 2017 2. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI, 2018 3. Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons, 2010 4. A Concise Introduction to Software Engineering, Pankaj Jalote, Springer 		

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		
List of sample projects <ol style="list-style-type: none"> a. Student Result Management System b. Library management system c. Inventory control system d. Accounting system e. Fast food billing system f. Bank loan system g. Blood bank system h. Railway reservation system i. Automatic teller machine j. Video library management system k. Hotel management system l. Hostel management system m. Share online trading n. Hostel management system o. Resource management system p. Court case management system 			

Course Code	Course Title	Credits	Lectures /Week
USCS404	IoT Technologies	2	3
About the Course:			
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.			
Course Objectives:			
<ul style="list-style-type: none"> ● Introduce concepts of SoC and IoT ● Introduce various types of IoT platforms ● Interfacing various types of devices using different protocols with IoT ● Understand practical applications of IoT in real life world 			
Learning Outcomes:			
After successful completion of this course, students would be able to			
<ul style="list-style-type: none"> ● understand SoC and IoT ● use different types of IoT Platforms and interfaces ● understand and implement an idea of various types of applications built using IoT 			
Unit	Topics	No of Lectures	
I	<p>Fundamentals of IoT: Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & 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>System on Chip: What is System on chip? Structure of System on Chip. SoC Elements: FPGA, GPU, APU, Compute Units.</p> <p>Different types of IoT/SoC Platforms: Introduction to Raspberry Pi, Arduino & NodeMCU, Introduction to SoC-ARM Architecture, atmega328 architecture</p>	15	
II	<p>Interfacing with IoT Platforms: 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>Using Sensor & Actuators: 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>IoT and Protocols IoT Security: HTTP, UPnP, CoAP, MQTT, XMPP, Privacy and Security Issues in IoT.</p>	15	

III	<p>IoT & Web: Web server for IoT, Sending/Receiving data between web server & IoT device, Cloud for IoT, Node RED, M2M vs IoT Communication Protocols, Basics of WSNs, WSN architecture and types,</p> <p>IoT Applications: Modern IoT case studies / applications used in the areas of transportation, agriculture, health care etc</p> <p>Edge Computing: 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>	15
------------	---	-----------

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

Course Code	Course Title	Credits	Lectures /Week
USCSP404	IoT Technologies – Practical	1	3
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • Kotlin Programming Language for application development • Creating robust mobile applications on simulators and physical devices • Creating intuitive, reliable mobile apps using the android services and components • Handling data local and remote data storage • Create a seamless user interface that works with different mobile screens 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Build useful mobile applications using Kotlin language on Android • Install and configure Android Studio for application development • Master basic to intermediate concepts of Kotlin required for mobile application development • Use built-in widgets and components, work with the database to store data • Master key Android programming concepts and deploy the application on Google Play 			
Unit	Topics	No of Lectures	
I	<p>Introduction to Kotlin: 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>Advanced Concepts in Kotlin: 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>App Development with Android Studio: 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>Basics Of Android- Application Components: Activities, Intent, and Broadcast Receiver, Services, Fragment, Activity Life Cycle, Content Provider, Widgets, and Notifications</p>	15	

<p style="text-align: center;">II</p>	<p>Designing Android UI: 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>Handle Images, Listview And Menu: ImageView, ImageSwitcher, ListView, Menu, and its types, Designing menu in XML, Option menu, Context menu, popup menu, Screen Navigation, RecyclerView, Interaction of Views</p> <p>Data binding in Android-AdapterView, Spinner, Gallery view, AutotextCompleteView, screen orientation, Design the view dynamically</p> <p>Implementing Data Persistence: 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;">15</p>
<p style="text-align: center;">III</p>	<p>Graphics, Animations, and Integrating Media in Android: Drawable Class, Animation in Android, MediaPlayer API and in Android, MediaPlayer and AudioManager Class,</p> <p>Interacting With Camera and input gestures: Android Camera, Input gestures-multiple touch, swipe, drag, scroll, zoom, Recording</p> <p>Gathering Location Data:</p> <p>Managing Background Tasks: Broadcast Receivers, Services, Threads and Process, AsyncTask, JobScheduler, Manage device Awake State</p> <p>Deploying Android applications on Google Play-Publishing/Deploy the application, Versioning, signing Application</p>	<p style="text-align: center;">15</p>
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. 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 2. Android Programming: Crafting UI/UX using Kotlin, SYBGEN Learning, 2020 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin Dawn Griffiths, 3rd Edition, O'Reilly Media, 2021 2. 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 3. Android Programming with Kotlin for Beginners, John Horton, Packt Publishing, 2019 4. Android Development with Kotlin: Enhance your skills for Android development using Kotlin, Marcin Moskala, Packt Publishing 		

Course Code	Course Title	Credits	Lectures /Week
USCSP405	Android Application Development – Practical	1	3
1	<ul style="list-style-type: none"> i. Write a program using Kotlin to implement control structures and loops. ii. Write a program to implement object-oriented concepts in Kotlin. 		
2	<ul style="list-style-type: none"> i. Create an Android application to design screens using different layouts and UI including Button, Edittext, Textview, Radio Button etc. ii. Write an android application demonstrating response to event/user interaction for <ul style="list-style-type: none"> a. Checkbox b. Radio button c. Button d. Spinner 		
3	<ul style="list-style-type: none"> i. Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image. ii. Create an application to use Gridview for shopping cart application. 		
4	<ul style="list-style-type: none"> i. Create an Android application to demonstrate implicit and explicit intents ii. Create an application to demonstrate shared preferences 		
5	<ul style="list-style-type: none"> i. Create an Android application to demonstrate the use of Broadcast listeners. ii. Create an Android application to create and use services. 		
6	<ul style="list-style-type: none"> i. Create an Android application to demonstrate XML based animation ii. Create an Android application to display canvas and allow the user to draw on it. 		
7	<ul style="list-style-type: none"> i. Create a media player application in android that plays audio. Implement play, pause, and loop features. ii. Create an Android application to use a camera and capture image/video and display them on the screen. 		
8	<ul style="list-style-type: none"> i. Create an android application to implement Asynctask and threading concepts. ii. Create an Android application to demonstrate the different types of menus. <ul style="list-style-type: none"> a. Pop-up Menu b. Context Menu c. Option Menu 		
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>About the Course: The course aims at developing scalable, robust, and maintainable web applications using MEAN stack and developing advanced mobile applications using Flutter</p>			
<p>Course Objectives:</p> <ul style="list-style-type: none"> To understand all the necessary and important technologies such as MongoDB, Express.js, AngularJS, and Node.js. To understand modern app development using Flutter 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> Store the data in NoSQL, document-oriented MongoDB database that brings performance and scalability. Use Node.js and Express Framework for building fast, scalable network applications Use AngularJS framework that offers declarative, two-way data binding for web applications. Integrate the front-end and back-end components of the MEAN stack. Develop robust mobile applications using Flutter. 			
Unit	Topics	No of Lectures	
I	<p>Node.js (N): 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>MongoDB(M): 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() & deleteMany()</p>	15	
II	<p>Server-Side Development with Express (E): 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>Understanding Angular.JS(A): 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.	
III	<p>Understanding Flutter: Importance of Flutter, Flutter Framework, Android Studio, Flutter SDK, Installing and Configuring Flutter SDK.</p> <p>Dart Programming: main() function, Dart Variables, Dart Data Types, Dart Conditional Operators, Control Flow & 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>Flutter Widgets Fundamentals: 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>Navigation and Routing: 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>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 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. 2. Beginning Flutter: A Hands On Guide to App Development by Marco L. Napoli, Wrox, 2019 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Full Stack Javascript Development with Mean - MongoDB, Express, AngularJS, and Node.JS by Adam Bretz, Colin J Ihrig, Shroff/SitePoint, 2015 2. Practical Flutter by Zammetti Frank, Apress, 2019 		

Course Code	Course Title	Credits	Lectures /Week
USCSP406	Advanced Application Development – Practical	1	3
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>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • The research methodology course is proposed to assist students in planning and carrying out research projects. • The students are exposed to the principles, procedures and techniques of implementing research project. • The course starts with an introduction to research and carries through the various methodologies involved. • It continues with finding out the literature using technology, basic statistics required for research and finally report writing. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Define research, formulate problem and describe the research process and research methods. • Understand and apply basic research methods including research design, data analysis and interpretation. • Understand ethical issues in research, write research report, research paper and publish the paper. 			
Unit	Topics	No of Lectures	
I	<p>Introduction to Research Methodology: 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>Defining the Research Problem: Concept and need, Identification of Research problem, defining and delimiting Research problem.</p> <p>Formulating a Research Problem: Reviewing Literature, formulating a Research Problem, Research Question, Identifying Variables, Constructing Hypothesis</p> <p>The Research Design: Meaning, Need for Research Design, Important Concepts, Different Research Designs, Basic Principles of Experimental Designs.</p>	15	
II	<p>Tools for Data Collection: 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>Sampling Design: 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>Overview of Hypothesis Testing: 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;">III</p>	<p>Technical Writing: Writing a Research Proposal, what is a Scientific Paper? Ethics in Scientific Publishing.</p> <p>Preparing the Text: 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>Preparing the Tables and Figures: How to Design Effective Tables, how to Prepare Effective Graphs, how to Prepare Effective Photographs.</p> <p>Publishing the Paper: 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;">15</p>
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Kothari C.R., Research Methodology, New Age International Publication, 2019 2. Research Methodology-A Step-by-Step Guide for Beginners, (4th ed.), Ranjit Kumar, Singapore, Pearson Education, 2018 3. Research Methodology, Vaishali Khairnar, Staredu Solutions India Pvt Ltd, 2020 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Research Methodology: Methods and Techniques, Dr. R. K. Jain, , Fifth Edition, VEI, 2021 2. Research Methodology, R. Panneerselvam, Second Edition, PHI, 2014 3. Dr. Rachna Jain, Research Methodology, Maximax Publishing House 4. How to Write and Publish a Scientific Paper, Cambridge University Press, Barbara Gastel and Robert A. Day, 2017 		

Course Code	Course Title	Credits	Lectures /Week
USCS4072	Management & Entrepreneurship	2	3
<p>About the Course: 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>Course Objectives:</p> <ul style="list-style-type: none"> • To understand the idea of management, process and its levels. • To understand the perception of entrepreneurship, process and its types. • To understand the concept SSI and steps to start SSI. • To understand the selection of project, project report, project appraisal, and its feasibility. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> • Understand the meaning of management, functions, administration and its process. • Understand the foundation of entrepreneurship and its theory, types and its process. • Identify the steps involved in an entrepreneurial venture (SSI). • Understand an entrepreneur is converting his business ideas into running concern by selecting the project. 			
Unit	Topics	No of Lectures	
I	<p>Introduction: Meaning, 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>Planning: Meaning, Nature, importance, types of planning, types of plans, planning process, decision-making.</p> <p>Organizing and staffing: 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>Directing and Controlling: 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>Entrepreneurship: 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>	
III	<p>Small Scale Industry: 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>Preparation of Project: 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>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Havinal Veerabhadrapa, Management and Entrepreneurship, New Age International Publishers. 2. Kanishka Bedi, Management and Entrepreneurship, Oxford University Press 3. Dr. R. K. Singal, Entrepreneurship Development and Management <p>Additional References:</p> <ol style="list-style-type: none"> 1. P. N. Singh, J. C. Saboo, Entrepreneurship Management, 6th Edition, Dr. P. N. Singh Centre for Hrd Publications. 2. Donald L. Sexton & Raymond W. Smilor, The Art and Science of Entrepreneurship, Ballinger, 2022 3. Clifford M. Baumback & Joseph R. Mancuso, Entrepreneurship And Venture Management, Prentice Hall 		

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:

All questions are compulsory.			
Question	Based on	Options	Marks
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.

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

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

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

USCS604	Information Retrieval	3	3
USCS605	Digital Image Processing	3	3
USCS606	Data Science	3	3
	Skill Enhancement		
USCS607	Ethical Hacking	2	3
	Practical		
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

Course: USCS501	TOPICS (Credits : 03 Lectures/Week:03) Artificial Intelligence	
Objectives: 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.		
Expected Learning Outcomes: 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.		
Unit I	What Is AI: Foundations, History and State of the Art of AI. Intelligent Agents: Agents and Environments, Nature of Environments, Structure of Agents. Problem Solving by searching: Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.	15L
Unit II	Learning from Examples: 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	15L

Unit III	Learning probabilistic models: 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.	15L
<p>Textbook(s):</p> <p>1) Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig,3rd Edition, Pearson, 2010.</p> <p>Additional Reference(s):</p> <p>1) Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017.</p> <p>2) Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017</p> <p>3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013</p>		

Course: USCS502	TOPICS (Credits : 03 Lectures/Week:03) Linux Server Administration
<p>Objectives:</p> <p>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.</p> <p>Expected Learning Outcomes:</p> <p>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.</p>	

Unit I	<p>Introduction: Technical Summary of Linux Distributions, Managing Software</p> <p>Single-Host Administration: Managing Users and Groups, Booting and shutting down processes, File Systems, Core System Services, Process of configuring, compiling, Linux Kernel</p> <p>Networking and Security: TCP/IP for System Administrators, basic network Configuration, Linux Firewall (Netfilter), System and network security</p>	15L
Unit II	<p>Internet Services: 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>	15L
Unit III	<p>Intranet Services: 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>	15L
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1) Linux Administration: A Beginner's Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016 2) Ubuntu Server Guide, Ubuntu Documentation Team, 2016 <p>Additional Reference(s):</p> <ol style="list-style-type: none"> 1) Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016 		

Course: USCS503	TOPICS (Credits : 03 Lectures/Week:03) Software Testing and Quality Assurance	
Objectives: <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> Expected Learning Outcomes: <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>		
Unit I	Software Testing and Introduction to quality : Introduction, Nature of errors, an example for Testing, Definition of Quality , QA, QC, QM and SQA , Software Development Life Cycle, Software Quality Factors Verification and Validation : Definition of V &V , Different types of V & V Mechanisms, Concepts of Software Reviews, Inspection and Walkthrough Software Testing Techniques : Testing Fundamentals, Test Case Design, White Box Testing and its types, Black Box Testing and its types	15L
Unit II	Software Testing Strategies : Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing Software Metrics : Concept and Developing Metrics, Different types of Metrics, Complexity metrics Defect Management: Definition of Defects, Defect Management Process, Defect Reporting, Metrics Related to Defects, Using Defects for Process Improvement.	15L
Unit III	Software Quality Assurance : 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	15L

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

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

Unit I	<p>Introduction: Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms</p> <p>Classical Encryption Techniques: 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>Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, The RSA Algorithm</p>	15L
Unit II	<p>Key Management: Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange</p> <p>Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC</p> <p>Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard</p> <p>Authentication Applications: Kerberos, X.509 Authentication, Public-Key Infrastructure</p>	15L
Unit III	<p>Electronic Mail Security: Pretty Good Privacy, S/MIME</p> <p>IP Security: Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management</p> <p>Web Security: Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction</p> <p>Intrusion: Intruders, Intrusion Techniques, Intrusion Detection</p> <p>Malicious Software: Viruses and Related Threats, Virus Countermeasures, DDOS</p> <p>Firewalls: Firewall Design Principles, Types of Firewalls</p>	15L
<p>Textbook(s):</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, 2nd Edition, TMH, 2011

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

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

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

Course: USCS506	TOPICS (Credits : 03 Lectures/Week:03) Web Services
<p>Objectives: 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>Expected Learning Outcomes: 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>	

Unit I	<p>Web services basics :</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>	15L
Unit II	<p>The REST Architectural style :</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>	15L
Unit III	<p>Developing Service-Oriented Applications with WCF :</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>	15L
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1) Web Services: Principles and Technology, Michael P. Papazoglou, Pearson Education Limited, 2008 2) RESTful Java Web Services, Jobinesh Purushothaman, PACKT Publishing, 2nd Edition, 2015 3) Developing Service-Oriented Applications with WCF, Microsoft, 2017 https://docs.microsoft.com/en-us/dotnet/framework/wcf/index <p>Additional Reference(s):</p> <ol style="list-style-type: none"> 1) Leonard Richardson and Sam Ruby, RESTful Web Services, O'Reilly, 2007 2) The Java EE 6Tutorial, Oracle, 2013 		

Course: USCS507	TOPICS (Credits : 03 Lectures/Week: 03) Game Programming	
Objectives: 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. Expected Learning Outcomes: 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.		
Unit I	Mathematics for Computer Graphics, DirectX Kickstart: Cartesian Coordinate system: 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 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 Transformations: 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 DirectX: Understanding GPU and GPU architectures. How they are different from CPU Architectures? Understanding how to solve by GPU?	15L

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

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, 2nd Edition, 1997
- 2) HLSL Development Cookbook, Doron Feinstein, PACKT Publishing,2013

Suggested List of Practical- SEMESTER V

Course: USCSP501	(Credits : 02 Lectures/Week: 06) Practical of Elective-I	
USCS501: Artificial Intelligence		
<p style="text-align: center;"><i>Practical shall be implemented in LISP</i></p> <ol style="list-style-type: none"> 1. Implement Breadth first search algorithm for Romanian map problem. 2. Implement Iterative deep depth first search for Romanian map problem. 3. Implement A* search algorithm for Romanian map problem. 4. Implement recursive best-first search algorithm for Romanian map problem. 5. Implement decision tree learning algorithm for the restaurant waiting problem. 6. Implement feed forward back propagation neural network learning algorithm for the restaurant waiting problem. 7. Implement Adaboost ensemble learning algorithm for the restaurant waiting problem. 8. Implement Naive Bayes' learning algorithm for the restaurant waiting problem. 9. Implement passive reinforcement learning algorithm based on adaptive dynamic programming (ADP) for the 3 by 4 world problem 10. Implement passive reinforcement learning algorithm based on temporal differences (TD) for 3 by 4 world problem. 		
USCS502: Linux Server Administration		
<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"> 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 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 3. Configure NTP 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 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.

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

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

Course: USCS601	TOPICS (Credits : 03 Lectures/Week: 03) Wireless Sensor Networks and Mobile Communication	
Objectives: 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.		
Expected Learning Outcomes: 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.		
Unit I	Introduction: 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. Sensor Node Hardware and Network Architecture: 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.	15L
Unit II	Medium Access Control Protocols: Fundamentals of MAC Protocols, MAC Protocols for WSNs, Sensor-MAC Case Study. Routing Protocols : Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless Sensor Networks, Routing Strategies in Wireless Sensor Networks. Transport Control Protocols : Traditional Transport Control Protocols,	15L

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

Course: USCS602	TOPICS (Credits : 03 Lectures/Week: 03) Cloud Computing	
<p>Objectives:</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>Expected Learning Outcomes:</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>		
Unit I	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.	15L
Unit II	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	15L
Unit III	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	15L

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>Unit II</p>	<p>Internet Forensic : Introduction 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 E-mail Forensics : e-mail analysis, e-mail headers and spoofing, Laws against e-mail Crime, Messenger Forensics: Yahoo Messenger Social Media Forensics: Social Media Investigations Browser Forensics: Cookie Storage and Analysis, Analyzing Cache and temporary internet files, Web browsing activity reconstruction</p>	<p>15L</p>
<p>Unit III</p>	<p>Investigation, Evidence presentation and Legal aspects of Digital Forensics: Authorization to collect the evidence , Acquisition of Evidence, Authentication of the evidence, Analysis of the evidence, Reporting on the findings, Testimony Introduction to Legal aspects of Digital Forensics: Laws & regulations, Information Technology Act, Giving Evidence in court, Case Study – Cyber Crime cases, Case Study – Cyber Crime cases</p>	<p>15L</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 Prosis, Tata McGrawHill,2nd Edition,2003

<p>Course: USCS604</p>	<p>TOPICS (Credits : 03 Lectures/Week: 03) Information Retrieval</p>	
<p>Objectives: 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>Expected Learning Outcomes:</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.

Unit I	Introduction to Information Retrieval: Introduction, History of IR, Components of IR, and Issues related to IR, Boolean retrieval, Dictionaries and tolerant retrieval.	15L
Unit II	Link Analysis and Specialized Search: 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.	15L
Unit III	Web Search Engine: Web search overview, web structure, the user, paid placement, search engine optimization/spam, Web size measurement, search engine optimization/spam, Web Search Architectures. XML retrieval: 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.	15L

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, 2nd Edition, ACM Press Books 2011.
- 3) Search Engines: Information Retrieval in Practice, Bruce Croft, Donald Metzler and Trevor Strohman, 1st 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)

Course: USCS605	TOPICS (Credits : 03 Lectures/Week: 03) Digital Image Processing	
<p>Objectives: 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>Expected Learning Outcomes: 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>		
Unit I	<p>Introduction to Image-processing System : Introduction, Image Sampling, Quantization, Resolution, Human Visual Systems, Elements of an Image-processing System, Applications of Digital Image Processing</p> <p>2D Signals and Systems : 2D signals, separable sequence, periodic sequence, 2D systems, classification of 2D systems, 2D Digital filter</p> <p>Convolution and Correlation : 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>Image Transforms: 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>	15L
Unit II	<p>Image Enhancement :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>	15L

	<p>Binary Image processing :Mathematical morphology, Structuring elements, Morphological image processing, Logical operations, Morphological operations, Dilation and Erosion, Distance Transform</p> <p>Colour Image processing :Colour images, Colour Model, Colour image quantization, Histogram of a colour image</p>	
Unit III	<p>Image Segmentation: Image segmentation techniques, Region approach, Clustering techniques, Thresholding, Edge-based segmentation, Edge detection, Edge Linking, Hough Transform</p> <p>Image Compression: 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>	15L
<p>Textbook(s):</p> <p>1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009</p> <p>Additional Reference(s):</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 (https://scilab.in/textbook_companion/generate_book/125)</p>		
Course: USCS606	<p>TOPICS (Credits : 03 Lectures/Week: 03)</p> <p>Data Science</p>	
<p>Objectives:</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>Expected Learning Outcomes:</p> <p>After completion of this course, the students should be able to understand & comprehend the problem; and should be able to define suitable statistical method to be adopted.</p>		
Unit I	Introduction to Data Science: What is Data? Different kinds of data,	15L

	<p>Introduction to high level programming language + Integrated Development Environment (IDE), Exploratory Data Analysis (EDA) + Data Visualization, Different types of data sources,</p> <p>Data Management: Data Collection, Data cleaning/extraction, Data analysis & Modeling</p>	
Unit II	<p>Data Curation: 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>	15L
Unit III	<p>Statistical Modelling and Machine Learning:</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>Data transformations: Dimension reduction, Feature extraction, Smoothing and aggregating</p> <p>Supervised Learning: Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN</p> <p>Unsupervised Learning: Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Ensemble methods</p>	15L
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1) Doing Data Science, Rachel Schutt and Cathy O’Neil, O’Reilly,2013 2) Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015 <p>Additional Reference(s):</p> <ol style="list-style-type: none"> 1) Hands-On Programming with R, Garrett Golemund,1st Edition, 2014 2) An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015 		

Course: USCS607	TOPICS (Credits : 02 Lectures/Week: 03) Ethical Hacking	
Objectives: To understand the ethics, legality, methodologies and techniques of hacking. Expected Learning Outcomes: 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.		
Unit I	Information Security : Attacks and Vulnerabilities Introduction to information security : Asset, Access Control, CIA, Authentication, Authorization, Risk, Threat, Vulnerability, Attack, Attack Surface, Malware, Security-Functionality-Ease of Use Triangle Types of malware : Worms, viruses, Trojans, Spyware, Rootkits Types of vulnerabilities : 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 Types of attacks and their common prevention mechanisms : 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 Case-studies : Recent attacks – Yahoo, Adult Friend Finder, eBay, Equifax, WannaCry, Target Stores, Uber, JP Morgan Chase, Bad Rabbit	15L
Unit II	Ethical Hacking – I (Introduction and pre-attack) Introduction: 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	15L

	<p>concerns, Black box vs. White box vs. Black box, Vulnerability assessment and Penetration Testing.</p> <p>Approach : Planning - 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>Enterprise strategy : Repeated PT, approval by security testing team, Continuous Application Security Testing,</p> <p>Phases: Reconnaissance/foot-printing/Enumeration, Phases: Scanning, Sniffing</p>	
<p>Unit III</p>	<p>Ethical Hacking :Enterprise Security</p> <p>Phases : Gaining and Maintaining Access : Systems hacking – 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, Applications hacking : SMTP/Email-based attacks, VOIP vulnerabilities, Directory traversal, Input Manipulation, Brute force attack, Unsecured login mechanisms, SQL injection, XSS, Mobile apps security, Malware analysis : Netcat Trojan, wrapping definition, reverse engineering</p> <p>Phases : Covering your tracks : Steganography, Event Logs alteration</p> <p>Additional Security Mechanisms : IDS/IPS, Honeypots and evasion techniques, Secure Code Reviews (Fortify tool, OWASP Secure Coding Guidelines)</p>	<p>15L</p>
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1) Certified Ethical Hacker Study Guide v9, Sean-Philip Oriyano, Sybex; Study Guide Edition,2016 2) CEH official Certified Ethical Hacking Review Guide, Wiley India Edition, 2007 <p>Additional Reference(s):</p>		

- 1) Certified Ethical Hacker: Michael Gregg, Pearson Education, 1st Edition, 2013
- 2) Certified Ethical Hacker: Matt Walker, TMH, 2011
- 3) http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines
- 4) 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
- 6) 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
- 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

Course: USCSP601	(Credits : 02 Lectures/Week:06) Practical of Elective-I	
USCS601: Wireless Sensor Networks and Mobile Communication		
<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"> 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. <ul style="list-style-type: none"> - nesC model - nesC Components 3. Understanding TOSSIM for <ul style="list-style-type: none"> - 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. 		
USCS602: Cloud Computing		
<ol style="list-style-type: none"> 1. Study and implementation of Infrastructure as a Service. 2. Installation and Configuration of virtualization using KVM. 3. Study and implementation of Infrastructure as a Service 4. Study and implementation of Storage as a Service 5. Study and implementation of identity management 6. Study Cloud Security management 		

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

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

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:

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

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

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

****Project Evaluation Scheme:**

Presentation	Working of the Project	Quality of the Project	Viva	Documentation
10Marks	10 Marks	10 Marks	10 Marks	10Marks

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