

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)	
Objectives. To understand the structure and operation of modern processors and their instruction sets Expected Learning Outcomes: <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	Computer Abstractions and Technology: Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters. 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. Sequential circuits and functions: State diagram and state table; finite state machines and their synthesis.	15 L
Unit II	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.	15 L
Unit III	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. Basic I/O: Accessing I/O devices, data transfers between processor and I/O devices. Interrupts and exceptions: interrupt requests and processing.	15 L

Text book:	
1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012	
Additional References:	
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. , 4 th Edition, 2010	

Course: USCS102	Programming with Python- I (Credits : 2 Lectures/Week: 3)	
Objectives The objective of this paper is to introduce various concepts of programming to the students using Python. Expected learning outcomes <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	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. 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. Interactive and script modes of IDLE, running a script, restarting the shell. 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).	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

	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.	
Unit II	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. Collaboration, Community and Communication Contributing to Open Source Projects Introduction to github, interacting with the community on github, Communication and etiquette, testing open source code, reporting 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.	15L
Unit III	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	15L
Text books: <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 Additional references: <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:</p> <p>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: <ol style="list-style-type: none"> 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>	

<p>USCSP102</p>	<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. 	
----------	--	--

USCSP103	<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 	
USCSP104	<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 	
USCSP105	<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. 	
USCSP106	<p>Descriptive Statistics and Introduction to Probability (To be implemented using R)</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)	
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.		
Expected Learning Outcomes <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	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(). 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 Variables: Declaring variables, scope of the variables according to block, hierarchy of data types. Compare explicit declarations in C with implicit declarations in Python. 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. 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.	15L
Unit II	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. Data Input and Output functions: Character I/O format: getch(), getche(), getchar(),getc(), gets(), putchar(), putc(), puts(). Manipulating Strings: Declaring and initializing String variables, Character and string handling functions. Compare with Python strings. Functions: Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values. Recursion: Definition, Recursive functions.	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)	
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.		
Expected Learning Outcomes <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) Students should know how to connect to computers, read from URL and send email. 		
Unit I	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. Exception handling: What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise. Regular Expressions: Concept of regular expression, various types of regular expressions, using match function.	15 L
Unit II	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	15 L
Unit III	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. Network connectivity: Socket module, creating server-client programs, sending email, reading from URL	15 L
Text books: <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 Additional References: <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)	
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. Expected Learning Outcomes: <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	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. Installation Installation methods, Hands on Installation using CD/DVD or USB drive. 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.	15L
Unit II	Graphical Desktop Session Management, Basic Desktop Operations, Network Management, Installing and Updating Software, Text editors: gedit, vi, vim, emacs, Graphics editors, Multimedia applications. Command Line Command line mode options, Shells, Basic Commands, General Purpose Utilities, Installing Software, User management, Environment variables, Command aliases. Linux Documentation man pages, GNU info, help command, More documentation sources File Operations Filesystem, Filesystem architecture, File types, File attributes, Working with files, Backup, compression	15L
Unit III	Security Understanding Linux Security, Uses of root, sudo command, working with passwords, Bypassing user authentication, Understanding ssh Networking Basic introduction to Networking, Network protocols: http, ftp etc., IP address, DNS, Browsers, Transferring files. ssh, telnet, ping, traceroute, route, hostname, networking GUI. 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	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	Recursion: Recursive Functions, Properties of Recursion, Its working, Recursive Applications Hash Table: Introduction, Hashing-Linear Probing, Clustering, Rehashing, Separate Chaining, Hash Functions Advanced Sorting: Merge Sort, Quick Sort, Radix Sort, Sorting Linked List Binary Trees: Tree Structure, Binary Tree-Properties, Implementation and Traversals, Expression Trees, Heaps and Heapsort, Search Trees	15L
Text book: <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 Additional References: <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)	
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.		
Expected Learning Outcomes: <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	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.	15L
Unit II	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.	15L
Unit III	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.	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: <ol style="list-style-type: none"> 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-Wallis tests. Post-hoc analysis of one-way analysis of variance : Duncan's test Chi-square test of association	15L
Text Book: <ol style="list-style-type: none"> 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: <ol style="list-style-type: none"> 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. (2002) : Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi 4. Gupta, S.C. and Kapoor, V.K. (4th 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.		

Expected Learning Outcomes: 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	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? Green Devices and Hardware: Introduction , Life Cycle of a Device or Hardware, Reuse, Recycle and Dispose Green Software: Introduction , Processor Power States , Energy-Saving Software Techniques, Evaluating and Measuring Software Impact to Platform Power Sustainable Software Development: Introduction, Current Practices, Sustainable Software, Software Sustainability Attributes, Software Sustainability Metrics, Sustainable Software Methodology, Defining Actions	15L
Unit II	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 Green Data Storage: Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management Green Networks and Communications: Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards 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.	15L
Unit III	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 Enterprise Green IT Readiness: Introduction, Readiness and Capability, Development of the G-Readiness Framework, Measuring an Organization's G-Readiness Sustainable IT Services: Creating a Framework for Service Innovation: Introduction, Factors Driving the Development of Sustainable IT, Sustainable IT Services (SITS), SITS Strategic Framework 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	15L
Text book: 1) <i>Harnessing Green IT: Principles and Practices</i> , San Murugesan, G. R. Ganadharan, Wiley & IEEE. Additional References: 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"> Linux Installation: <ol style="list-style-type: none"> Install your choice of Linux distribution e.g. Ubuntu, Fedora, Debian. Try different installation media like CD/DVD, USB Drive to install. Customize desktop environment by changing different default options like changing default background, themes, screensavers. <ol style="list-style-type: none"> Screen Resolution: Ascertain the current screen resolution for your desktop. 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? 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. Installing and Removing Software: <ol style="list-style-type: none"> Install gcc package. Verify that it runs, and then remove it. Documentations: <ol style="list-style-type: none"> Finding Info Documentation: From the command line: bring up the info page for the grep command. Bring up the usage section. Finding man pages From the command line: Bring up the man page for the 'ls' command. Scroll down to the EXAMPLES section. Finding man pages by Topic What man pages are available that document file compression? 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? Command-Line Help List the available options for the mkdir command. How can you do this? Command line operations: <ol style="list-style-type: none"> Install any newpackage on your system Remove the package installed Find the passwd file in / using find command Create a symbolic link to the file you found in last step Create an empty file example.txt and move it in /tmp directory using relative pathname. Delete the file moved to /tmp in previous step using absolute path. Find the location of ls, ps, bash commands. File Operations: <ol style="list-style-type: none"> Explore mounted filesystems on your system. What are different ways of exploring mounted filesystems on Linux? Archive and backup your home directory or work directory using tar, gzip commands. Use dd command to create files and explore different options to dd. Use diff command to create diff of two files. 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: vim/emacs <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>	
--	---	--

USCSP204	Data structures <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 	
USCSP205	Calculus <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 	
USCSP206	Statistical Methods and Testing of Hypothesis <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 	

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

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

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

SEMESTER III

THEORY

Course: USCS301	TOPICS (Credits : 02 Lectures/Week:03) Theory of Computation	
Objectives: To provide the comprehensive insight into theory of computation by understanding grammar, languages and other elements of modern language design. Also to develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas. Expected Learning Outcomes: <div><div>1. Understand Grammar and Languages</div><div>2. Learn about Automata theory and its application in Language Design</div><div>3. Learn about Turing Machines and Pushdown Automata</div><div>4. Understand Linear Bound Automata and its applications</div></div>		
Unit I	Automata Theory: Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata. Formal Languages: Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata	15L
Unit II	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 Context Free Languages: Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG Pushdown Automata: Definitions, Acceptance by PDA, PDA and CFG	15L

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

Course: USCS302	TOPICS (Credits : 02 Lectures/Week:03) Core Java	
Objectives: The objective of this course is to teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and to cover-up with the pre-requisites of Core java.		
Expected Learning Outcomes: <div><div></div><div>1. Object oriented programming concepts using Java.</div><div>2. Knowledge of input, its processing and getting suitable output.</div><div>3. Understand, design, implement and evaluate classes and applets.</div><div>4. Knowledge and implementation of AWT package.</div></div>		
Unit I	The Java Language: Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays OOPS: Introduction, Class, Object, Static Keywords, Constructors, this Key Word, Inheritance, super Key Word, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces String Manipulations: String, String Buffer, String Tokenizer Packages: Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers	15L
Unit II	Exception Handling: Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples Multithreading: Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods I/O Streams: Introduction, Byte-oriented streams, Character- oriented streams, File, Random access File, Serialization Networking: Introduction, Socket, Server socket, Client –Server Communication	15L
	Wrapper Classes: Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes Collection Framework: Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes	

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

Textbook(s):

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

Additional Reference(s):

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

Course: USCS303	TOPICS (Credits : 02 Lectures/Week:03) Operating System	
Objectives: Learners must understand proper working of operating system. To provide a sound understanding of Computer operating system, its structures, functioning and algorithms.		
Expected Learning Outcomes: 1. To provide a understanding of operating system, its structures and functioning 2. Develop and master understanding of algorithms used by operating systems for various purposes.		
Unit I	Introduction and Operating-Systems Structures: Definition of Operating system, Operating System’s role, Operating-System Operations, Functions of Operating System, Computing Environments Operating-System Structures: Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure Processes: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication	15L

	Threads: Overview, Multicore Programming, Multithreading Models	
Unit 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> <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>	15L
Unit 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>	15L
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 8th Edition <p>Additional Reference(s):</p> <ol style="list-style-type: none"> 1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill 2. Naresh Chauhan, Principles of Operating Systems, Oxford Press 3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016 		

Course: USCS304	TOPICS (Credits : 02 Lectures/Week:03) Database Management Systems	
Objectives: To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage. Expected Learning Outcomes: <div><div>1. Master concepts of stored procedure and triggers and its use.</div><div>2. Learn about using PL/SQL for data management</div><div>3. Understand concepts and implementations of transaction management and crash recovery</div></div>		
Unit I	Stored Procedures: Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. Triggers: Concept of triggers, Implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers. Sequences: creating sequences, referencing, altering and dropping a sequence. File Organization and Indexing: Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes.	15L
	Fundamentals of PL/SQL: Defining variables and constants, PL/SQL expressions and comparisons: Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements, PL/SQL Datatypes: Number Types, Character Types, Boolean Type, Datetime and Interval Types.	

Unit II	Overview of PL/SQL Control Structures: Conditional Control: IF and CASE Statements, IF-THEN Statement, IF-THEN-ELSE Statement, IFTHEN-ELSIF Statement, CASE Statement, Iterative Control: LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP, Sequential Control: GOTO and NULL Statements	15L
Unit III	<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>DCL Statements: Defining a transaction, Making Changes Permanent with COMMIT, Undoing Changes with ROLLBACK, Undoing Partial Changes with SAVEPOINT and ROLLBACK</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>	15L
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1) Ramakrishnam, Gehrke, Database Management Systems, Bayross, McGraw-Hill, 3rd Edition 2) Abraham Silberschatz, Henry F. Korth, S. Sudarshan , Database System Concepts, 6th Edition 3) Ivan Bayross, “SQL, PL/SQL -The Programming language of Oracle”, B.P.B. Publications <p>Additional Reference(s):</p> <ol style="list-style-type: none"> 1) Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education 2) Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press. 3) Joel Murach, Murach’s MySQL, Murach 		

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

	A Concrete Example, Integer Solutions of Linear Programming Problems. Combinatorial Applications of Network Flows: Introduction, Matching in Bipartite Graphs, Chain partitioning, Pólya's Enumeration Theorem: Coloring the Vertices of a Square.	
Textbook(s): 1) Applied Combinatorics, Mitchel T. Keller and William T. Trotter, 2016, http://www.rellek.net/appcomb .		
Additional Reference(s): 1) Applied Combinatorics, sixth.edition, Alan Tucker, Wiley; (2016) 2) Graph Theory and Combinatorics, Ralph P. Grimaldi, Pearson Education; Fifth edition (2012) 3) Combinatorics and Graph Theory, John Harris, Jeffrey L. Hirst, Springer(2010). 4) Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008).		

Course: USCS306	TOPICS (Credits : 02 Lectures/Week:03) Physical Computing and IoT Programming
Objectives: To learn about SoC architectures; Learn how Raspberry Pi. Learn to program Raspberry Pi. Implementation of internet of Things and Protocols.	
Expected Learning Outcomes: 1. Enable learners to understand System On Chip Architectures. 2. Introduction and preparing Raspberry Pi with hardware and installation. 3. Learn physical interfaces and electronics of Raspberry Pi and program them using practical's 4. Learn how to make consumer grade IoT safe and secure with proper use of protocols.	

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

Course: USCS307	TOPICS (Credits : 02 Lectures/Week: 03) Web Programming	
Objectives: To provide insight into emerging technologies to design and develop state of - the art web applications using client-side scripting, server-side scripting, and database connectivity.		
Expected Learning Outcomes: <ol style="list-style-type: none"> 1. To design valid, well-formed, scalable, and meaningful pages using emerging technologies. 2. Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites 3. To develop and implement client-side and server-side scripting language programs. 4. To develop and implement Database Driven Websites. 5. Design and apply XML to create a markup language for data and document centric applications. 		
Unit I	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 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	15L
Unit II	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 XML: Comparing XML with HTML, Advantages and Disadvantages of XML,	15L

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

Suggested List of Practical- SEMESTER III

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

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

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

3. Synchronization:

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

- 4. Implement FCFS scheduling algorithm in Java.
- 5. Implement SJF (with no preemption) scheduling algorithm in Java
- 6. Implement RR scheduling algorithm in Java
- 7. Write a Java program that implements the banker's algorithm
- 8. Write a Java program that implements the FIFO page-replacement algorithm.
- 9. Write a Java program that implements the LRU page-replacement algorithm.
- 10. Design a File System in Java.

USCS304: Database Management Systems

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

Course: USCSP302	(Credits : 03 Lectures/Week: 09) USCS305+ USCS306+USCS307	
USCS305: Combinatorics and Graph Theory		
<ol style="list-style-type: none"> 1. Solving problems on strings, sets and binomial coefficients. 2. Solving problems using induction. 3. Solving problems on Eulerian and Hamiltonian graphs. 4. Solving problems on Chromatic number and coloring 5. Solving problems using Kruskal's Algorithm 6. Solving problems using Prim's Algorithm 7. Solving problems using Dijkstra's Algorithm 8. Solving problems of finding augmenting paths in network flows. 9. Solving problems on network flows using Ford-Fulkerson Labeling Algorithm 10. Solving problems on posets and their associated networks. 		
USCS306: Physical Computing and IoT Programming		
<ol style="list-style-type: none"> 1. Preparing Raspberry Pi: Hardware preparation and Installation 2. Linux Commands: Exploring the Raspbian 3. GPIO: Light the LED with Python 4. GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas 5. SPI: Camera Connection and capturing Images using SPI 6. Real Time Clock display using PWM. 7. Stepper Motor Control: PWM to manage stepper motor speed. 8. Node RED: Connect LED to Internet of Things 9. Stack of Raspberry Pi for better Computing and analysis 10. Create a simple Web server using Raspberry Pi 		
USCS307: Web Programming		
<ol style="list-style-type: none"> 1. Design a webpage that makes use of <ol 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 <ol 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
 - a. CSS properties to change the background of a Page
 - b. CSS properties to change Fonts and Text Styles
 - c. CSS properties for positioning an element
4. Write JavaScript code for
 - a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
 - b. Validating the various Form Elements
5. Write JavaScript code for
 - a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date
 - b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document,
 - c. Storing and Retrieving Cookies
6. Create a XML file with Internal / External DTD and display it using
 - a. CSS
 - b. XSL
7. Design a webpage to handle asynchronous requests using AJAX on
 - a. Mouseover
 - b. button click
8. Write PHP scripts for
 - a. Retrieving data from HTML forms
 - b. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number
 - c. Working with Arrays
 - d. Working with Files (Reading / Writing)
9. Write PHP scripts for
 - a. Working with Databases (Storing Records / Reprieving Records and Display them)
 - b. Storing and Retrieving Cookies
 - c. Storing and Retrieving Sessions
10. Design a webpage with some jQuery animation effects.

SEMESTER IV

THEORY

Course: USCS401	TOPICS (Credits : 02 Lectures/Week:03) Fundamentals of Algorithms	
Objectives: <ol style="list-style-type: none">1. To understand basic principles of algorithm design and why algorithm analysis is important2. To understand how to implement algorithms in Python3. To understand how to transform new problems into algorithmic problems with efficient solutions4. To understand algorithm design techniques for solving different problems Expected Learning Outcomes: <ol style="list-style-type: none">1. Understand the concepts of algorithms for designing good program2. Implement algorithms using Python		
Unit I	Introduction to algorithm, Why to analysis algorithm, Running time analysis, How to Compare Algorithms, Rate of Growth, Commonly Used Rates of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega- Ω Notation, Theta- Θ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Performance characteristics of algorithms, Master Theorem for Divide and Conquer, Divide and Conquer Master Theorem: Problems & Solutions, Master Theorem for Subtract and Conquer Recurrences, Method of Guessing and Confirming	15L
Unit II	Tree algorithms: What is a Tree? Glossary, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals, Expression Trees, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees Graph Algorithms: Introduction, Glossary, Applications of Graphs, Graph Representation, Graph Traversals, Topological Sort, Shortest Path Algorithms, Minimal Spanning Tree	15L

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

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

Textbook(s):

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

Additional Reference(s):

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

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

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

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

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

	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	
Unit II	<p>System Design: System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design</p> <p>Software Measurement and Metrics: Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality</p> <p>Software Project Management: Estimation in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision, Project Scheduling - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts</p>	15L
Unit III	<p>Risk Management - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan</p> <p>Software Quality Assurance: Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, Software Reliability, The ISO 9000 Quality Standards, Capability Maturity Model</p> <p>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>	15L

Text book(s):

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

Additional Reference(s):

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

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

	<p>From function inverse to matrix inverse</p> <p>Basis: Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Linear dependence, Basis , Unique representation, Change of basis, first look, Computational problems involving finding a basis</p> <p>Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator</p>	
Unit III	<p>Gaussian elimination: Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers,</p> <p>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>Eigenvector: Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and eigenvectors, Coordinate representation in terms of eigenvectors, The Internet worm, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.</p>	15L
<p>Textbook(s):</p> <ol style="list-style-type: none"> 1) Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013) <p>Additional References:</p> <ol style="list-style-type: none"> 1) Linear Algebra and Probability for Computer Science Applications, 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) 		

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

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

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

Expected Learning Outcomes:

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

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

Textbook(s):

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

Additional Reference(s):

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

Suggested List of Practical – SEMESTER IV

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

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

USCS403: Computer Networks

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

Course: USCSP402	(Credits : 03 Lectures/Week:09) USCS405+ USCS406+ USCS407	
USCS405: Linear Algebra using Python		
<ol style="list-style-type: none"> 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. 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 Write a program to do the following: <ul style="list-style-type: none"> Enter two distinct faces as vectors u and v. Find a new face as a linear combination of u and v i.e. $au+bv$ for a and b in R. Find the average face of the original faces. Write a program to do the following: <ul style="list-style-type: none"> Enter an r by c matrix M (r and c being positive integers) Display M in matrix format Display the rows and columns of the matrix M Find the scalar multiplication of M for a given scalar. Find the transpose of the matrix M. Write a program to do the following: <ul style="list-style-type: none"> Find the vector –matrix multiplication of a r by c matrix M with an c-vector u. Find the matrix-matrix product of M with a c by p matrix N. Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse. Write a program to convert a matrix into its row echelon form. 		

8. Write a program to do the following:

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

9. Write a program to do the following:

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

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

USCS406: .NET Technologies

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

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

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

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

3. Design ASP.NET Pages with

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

4. Design ASP.NET Pages for State Management using

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

5. Perform the following activities

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

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

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

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

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

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

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

USCS407:Android Developer Fundamentals
<ol style="list-style-type: none">1. Install Android Studio and Run Hello World Program.2. Create an android app with Interactive User Interface using Layouts.3. Create an android app that demonstrates working with TextView Elements.4. Create an android app that demonstrates Activity Lifecycle and Instance State.5. Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers.6. Create an android app that demonstrates the use of an Options Menu.7. Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs.8. Create an android app to Connect to the Internet and use BroadcastReceiver.9. Create an android app to show Notifications and Alarm manager.10. Create an android app to save user data in a database and use of different queries.

T.Y.B.Sc. (Semester V and VI)
Computer Science Syllabus
Choice Based Credit 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
Textbook(s): 1) Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010. Additional Reference(s): 1) Artificial Intelligence: Foundations of Computational Agents, David L Poole, Alan K. Mackworth, 2nd Edition, Cambridge University Press, 2017. 2) Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017 3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013		

Course: USCS502	TOPICS (Credits : 03 Lectures/Week:03) Linux Server Administration
Objectives: Demonstrate proficiency with the Linux command line interface, directory & file management techniques, file system organization, and tools commonly found on most Linux distributions. Effectively operate a Linux system inside of a network environment to integrate with existing service solutions. Demonstrate the ability to troubleshoot challenging technical problems typically encountered when operating and administering Linux systems. Expected Learning Outcomes: Learner will be able to develop Linux based systems and maintain. Learner will be able to install appropriate service on Linux server as per requirement. Learner will have proficiency in Linux server administration.	

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: 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. To provide skills to design test case plan for testing software		
Expected Learning Outcomes: 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.		
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 		

Course:	TOPICS (Credits : 03 Lectures/Week:03)
USCS504	Information and Network Security
<p>Objectives:</p> <p>To provide students with knowledge of basic concepts of computer security including network security and cryptography.</p> <p>Expected Learning Outcomes:</p> <p>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

Course: USCS505	TOPICS (Credits : 03 Lectures/Week:03) Architecting of IoT	
Objectives: Discovering the interconnection and integration of the physical world. Learner should get knowledge of the architecture of IoT. 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.		
Unit I	IoT-An Architectural Overview: Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. 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	15L
Unit II	IoT Data Link Layer and Network Layer Protocols: PHY/MAC Layer(3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART,Z-Wave, Bluetooth Low Energy, Zigbee Smart Energy DASH7 Network Layer: IPv4, IPv6, 6LoWPAN, 6TiSCH,ND, DHCP, ICMP, RPL, CORPL, CARP	15L

Unit III	Transport layer protocols : Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP)-(TLS, DTLS) Session layer: Session Layer-HTTP, CoAP, XMPP, AMQP, MQTT Service layer protocols: Service Layer -oneM2M, ETSI M2M, OMA, BBF	15L
Textbook(s): <ol style="list-style-type: none"> 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): <ol style="list-style-type: none"> 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-onApproach), Vijay Madisetti and ArshdeepBahga,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
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	
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	

Unit I	Web services basics : 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	15L
Unit II	The REST Architectural style : 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	15L
Unit III	Developing Service-Oriented Applications with WCF : 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	15L
Textbook(s): 1) Web Services: Principles and Technology, Michael P. Papazoglou, Pearson Education Limited, 2008 2) RESTful Java Web Services, Jobinesh Purushothaman, PACKT Publishing, 2 nd Edition, 2015 3) Developing Service-Oriented Applications with WCF, Microsoft, 2017 https://docs.microsoft.com/en-us/dotnet/framework/wcf/index Additional Reference(s): 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 gamming 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

Unit II	<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>	15L
Unit III	<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>	15L
<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><i>Practical shall be implemented in Python</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

<p>CRUD operation.</p> <p>8. Implement a typical service and a typical client using WCF.</p> <p>9. Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service.</p> <p>10. Demonstrates using the binding attribute of an endpoint element in WCF.</p>		
<p>Course: USCSP503</p>	<p>(Credits : 01 Lectures/Week: 03) Project Implementation</p>	
<p><i>Please Refer to Project Implementation Guidelines</i></p>		
<p>Course: USCSP504</p>	<p>(Credits : 01 Lectures/Week: 03) Practical of Skill Enhancement</p>	
<p>USCS507 : Game Programming</p>		
<p>1. Setup DirectX 11, Window Framework and Initialize Direct3D Device</p> <p>2. Buffers, Shaders and HLSL (Draw a triangle using Direct3D 11)</p> <p>3. Texturing (Texture the Triangle using Direct 3D 11)</p> <p>4. Lightning (Programmable Diffuse Lightning using Direct3D 11)</p> <p>5. Specular Lightning (Programmable Spot Lightning using Direct3D 11)</p> <p>6. Loading models into DirectX 11 and rendering.</p> <p>Perform following Practical using online content from the Unity Tutorials Web--sites: https://unity3d.com/learn/tutorials/s/interactive-tutorials</p> <p>7. https://unity3d.com/learn/tutorials/s/2d-ufo-tutorial</p> <p>8. https://unity3d.com/learn/tutorials/s/space-shooter-tutorial</p> <p>9. https://unity3d.com/learn/tutorials/s/roll-ball-tutorial</p> <p>10. https://unity3d.com/learn/tutorials/topics/vr/introduction?playlist=22946</p>		

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, Waltenegus 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	
Objectives: 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.		
Expected Learning Outcomes: 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.		
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

Unit II	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	15L
Unit III	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	15L
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 Prosise, Tata McGrawHill, 2 nd Edition, 2003		

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

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	
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.		
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.		
Unit I	Introduction to Image-processing System : Introduction, Image Sampling, Quantization, Resolution, Human Visual Systems, Elements of an Image-processing System, Applications of Digital Image Processing 2D Signals and Systems : 2D signals, separable sequence, periodic sequence, 2D systems, classification of 2D systems, 2D Digital filter 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 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	15L
Unit II	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	15L

	Binary Image processing :Mathematical morphology, Structuring elements, Morphological image processing, Logical operations, Morphological operations, Dilation and Erosion, Distance Transform Colour Image processing :Colour images, Colour Model, Colour image quantization, Histogram of a colour image	
Unit III	Image Segmentation: Image segmentation techniques, Region approach, Clustering techniques, Thresholding, Edge-based segmentation, Edge detection, Edge Linking, Hough Transform 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	15L
Textbook(s): 1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009 Additional Reference(s): 1) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008 2) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan And T. Veerakumar, 2016 (https://scilab.in/textbook_companion/generate_book/125)		
Course: USCS606	TOPICS (Credits : 03 Lectures/Week: 03) Data Science	
Objectives: 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. Expected Learning Outcomes: 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.		
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>	
Unit III	<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>	15L
<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: USCSP602	(Credits : 02 Lectures/Week:06) Practical of Elective-II	
USCS604: Information Retrieval		
<p><i>Practical may be done using software/tools like Python / Java / Hadoop</i></p> <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	
<i>Please Refer to Project Implementation Guidelines</i>		
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)