(2)

UNIVERSITY OF MUMBAI
No. UG/ 107 of 2017

CIRCULAR:-

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

25/7/17

MUMBAI – 400 032 REGISTRAR
27th July, 2017

To,

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

A.C/4.210/11.05.2017

No. UG/107-A of 2017 MUMBAI-400 032 27th July, 2017

Copy forwarded with compliments for information to :-

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

25/7/17

REGISTRAR

....PTO

AC_____

UNIVERSITY OF MUMBAI

Syllabus for SemIII& IV
Program: B.Sc.
Course: Computer Science

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

Preamble

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

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

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

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

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

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

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

<table>
<tr><td>Course:
USCS301</td><td colspan="2">TOPICS (Credits : 02 Lectures/Week:03)
Theory of Computation</td></tr>
<tr><td colspan="3">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:
1. Understand Grammar and Languages
2. Learn about Automata theory and its application in Language Design
3. Learn about Turing Machines and Pushdown Automata
4. Understand Linear Bound Automata and its applications</td></tr>
<tr><td>Unit I</td><td>Automata Theory: Defining Automaton, Finite Automaton, Transitios and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata.
Formal Languges: Defining Grammar, Derivations, Languges generated by Grammar, Comsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata</td><td>15L</td></tr>
<tr><td>Unit II</td><td>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</td><td>15L</td></tr>
</table>

Unit III	**Linear Bound Automata:** The Linear Bound Automata Model, Linear Bound Automata and Languages. **Turing Machines:** Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine, **Undecidability:** The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems	**15L**

Tutorials :

1. Problems on generating languages for given simple grammar
2. Problems on DFA and NDFA 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

Textbook(s):

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.

Additional Reference(s):

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:** 1. Object oriented programming concepts using Java. 2. Knowledge of input, its processing and getting suitable output. 3. Understand, design, implement and evaluate classes and applets. 4. Knowledge and implementation of AWT package.		
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	**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 **CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	**15L**
Unit III	**Main Memory:** Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing **Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management **File-System Interface:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing **File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management	**15L**

Textbook(s):

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley,8th Edition

Additional Reference(s):

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:** 1. Master concepts of stored procedure and triggers and its use. 2. Learn about using PL/SQL for data management 3. Understand concepts and implementations of transaction management and crash recovery		
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	**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. **DCL Statements:** Defining a transaction, Making Changes Permanent with COMMIT, Undoing Changes with ROLLBACK, Undoing Partial Changes with SAVEPOINT and ROLLBACK **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.	**15L**

Textbook(s):

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

Additional Reference(s):

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:

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, Jeffry L. Hirst, Springer(2010).
4) Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008).

Course:	**TOPICS (Credits : 02 Lectures/Week:03)**
USCS306	**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:

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):

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):

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

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

Practical can be implemented either in JAVA or any other programming language.

1. **Process Communication**:
   (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**:
   (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 | NOORECER.
   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

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

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

1. Design a webpage that makes use of
   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
   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: 1. To understand basic principles of algorithm design and why algorithm analysis is important 2. To understand how to implement algorithms in Python 3. To understand how to transform new problems into algorithmic problems with efficient solutions 4. To understand algorithm design techniques for solving different problems **Expected Learning Outcomes:** 1. Understand the concepts of algorithms for designing good program 2. 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	Algorithms Design Techniques: Introduction, Classification, Classification by Implementation Method, Classification by Design Method Greedy Algorithms: Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and Disadvantages of Greedy Method, Greedy Applications, Understanding Greedy Technique 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 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	**15L**

Textbook(s):

1. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi , CareerMonk Publications, 2016
2. Introduction to Algorithm, Thomas H Cormen, PHI

Additional References(s):

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:** 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: 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. **Expected Learning Outcome**s : 1. Learner will be able to understand the concepts of networking, which are important for them to be known as a '*networking professionals*'. 2. Useful to proceed with industrial requirements and International vendor certifications.		
Unit I	**Introduction Network Models:** 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. Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture, Layers in	**15L**

	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. 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)	
Unit II	**Introduction to Physical Layer and Data-Link Layer:** 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, 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,	**15L**
Unit III	**Network layer, Transport Layer** Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA. Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer	**15L**

	Switches, Routers, Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing. 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. Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.	

Textbook(s):

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.

Additional Reference(s)*:*

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	**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 **Requirement Analysis and System Modeling:** Requirements Engineering, Eliciting Requirements, SRS Validation, Components of	**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	**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 **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 **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	**15L**
Unit III	**Risk Management** - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan **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 **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	**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**: 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**

	From function inverse to matrix inverse **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 Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator	
Unit III	**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, **Inner Product**: The inner product for vectors over the reals, Orthogonality, **Orthogonalization**: Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement, 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.	**15L**

Textbook(s):

1) Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)

Additional References*:*

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:** 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

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

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

1. Write a program which demonstrates the following:
    - Addition of two complex numbers
    - Displaying the conjugate of a complex number
    - Plotting a set of complex numbers
    - Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number a=1/2, a=1/3, a=2 etc.
2. Write a program to do the following:
    - Enter a vector u as a n-list
    - Enter another vector v as a n-list
    - Find the vector au+bv for different values of a and b
    - Find the dot product of u and v
3. Write a program to do the following:
    - 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.
4. Write a program to do the following:
    - 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.
5. Write a program to do the following:
    - 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.
6. Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse.
7. 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

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

Evaluation Scheme

I. Internal Exam - 25 Marks

(i) Test – 20 Marks

20 marks Test – Duration 40 mins
It will be conducted either using any open source learning management system like Moodle (Modular object-oriented dynamic learning environment)

OR

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

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

II. External Exam– 75 Marks

III. Practical Exam – 50 Marks

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

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

****