Shri Vile Parle Kelavani Mandal's

MITHIBAI COLLEGE OF ARTS, CHAUHAN INSTITUTE OF SCIENCE &
AMRUTBEN JIVANLAL COLLEGE OF COMMERCE AND ECONOMICS
(AUTONOMOUS)

NAAC Reaccredited 'A' grade, CGPA: 3.57 (February 2016),
Granted under RUSA, FIST-DST & -Star College Scheme of DBT, Government of Indic
Best College (2016-17), University of Mumbai

Affiliated to the

UNIVERSITY OF MUMBAI

Program: S.Y.B.Sc.

Course: Computer Science

Semester: III & IV

Choice Based Credit System (CBCS) with effect from the

Academic year 2022-23

A.C. No: 12

Agenda No: 4.4

Neelam Jain

Ashish Garande

Dr. Amol Joglekar

Jayshree Ravi

Zahir J. Mulani

PROGRAMME SPECIFIC OUTCOMES (PSO'S)

On completion of the B.Sc Computer Science, the learners should be enriched with knowledge and be able to-

PS01: To train the students for software development using different programming languages.
PS02: To develop the skills for problem solving in computing and other relevant disciplines.
PS03: To introduce emerging trends to the students in a gradual way.
PS04: To groom the students for facing the challenges in ICT industry.

Preamble

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

The B.Sc. Computer Science course structure therefore needed a fresh outlook and complete overhaul. A real genuine attempt has been made while designing the new syllabus for this three year graduate course. Not only does it prepare the students for a career in Software industry, it also motivates them towards further studies and research opportunities. In the third year i.e. for semester III & IV, learners will be taught theoretical concepts of Computer Science. The syllabus proposes to have six core subjects of Computer science and one core course of Mathematics. All core subjects except one are proposed to have theory as well as practical tracks. While the Computer Science courses will built theoretical background in Computer Science, the Mathematics course will inculcate research oriented acumen. The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science. We sincerely believe that any student taking this course will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students' community and teachers' fraternity will appreciate the treatment given to the courses in the syllabus.

The courses are as follows: -

Semester – III		
Course Title	**Credits**	**Lecture/Week**
Theory of Computation	2	2
Java Programming	2	2
Operating System	2	2
Database Management System	2	2
Advanced Web Technologies	2	2
Computer Networks	2	2
Combinatorics and Graph Theory	2	2
Computer Science Practical – VII	3	6
Computer Science Practical – VIII	3	6

Semester – IV		
Course Title	**Credits**	**Lecture/Week**
Fundamentals of Algorithms	2	2
Advanced Java	2	2
Physical Computing and IoT Programming	2	2
Android Developer Fundamentals	2	2
Linear Algebra with Python	2	2
.NET Technologies	2	2
Software Engineering	2	2
Computer Science Practical – IX	3	6
Computer Science Practical – X	3	6

N.B.- (i) The duration of each theory lecture will be of 60 minutes. A course consists of 3 modules. For each module the number of hours allotted are 10. The total number of lecture hours for each course will thus be 30.

For theory component value of One Credit is equal to 15 learning hours.

(ii) There will be one practical per batch for all but one courses per semester. The duration of each practical will be of 2 hours.

For practical component the value of One Credit is equal to 30 learning hours.

Thus in a week, a student will study 14 hours of theory and 12 hours of practical for each semester.

Evaluation Pattern

The performance of the learner will be evaluated in two components. The first component will be a Continuous Assessment with a weightage of 25% of total marks per course. The second component will be a Semester end Examination with a weightage of 75% of the total marks per course. The allocation of marks for the Continuous Assessment and Semester end Examinations is as shown below:

a) **Details of Continuous Assessment (CA)**

25% of the total marks per course:

Continuous Assessment	**Details**	**Marks**
Component 1 (CA-1)	Class Test/Research Paper Review/ Assignment/ Presentation/ Mini Project	15 marks
Component 2 (CA-2)	Assignment/ presentation/mini project, etc	10 marks

Minimum 2 component of Continuous Assessment need to be conducted per course.

b) **Details of Semester End Examination**

75% of the total marks per course. Duration of examination will be two and half hours.

Question Number	**Description**	**Marks**	**Total Marks**
Q.1	Answer any 3 out of 4 Questions	7 marks each	3 x 7 = 21 marks
Q.2	Answer any 3 out of 4 Questions	7 marks each	3 x 7 = 21 marks
Q.3	Answer any 3 out of 4 Questions	7 marks each	3 x 7 = 21 marks
Q.4	Answer any 3 out of 4 questions	4 marks each	3 x 4 = 12 marks
		Total Marks	**75 marks**

Evaluation for practical papers

In the Practical exams, there will be 20% assessment for the journal and laboratory work and 80% as term end component to be conducted as a semester end exam per course. For each course there will be one examiner per batch who will evaluate the practical.

Signature

HOD

mVaidya

Signature

Approved by Vice –Principal

Signature

Approved by Principal

Program: B.Sc.- Computer Science				**Semester : III**	
Course: Theory of Computation				**Course Code: USMACS301**	
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	-	-	2	25	75

Learning Objectives:

- To provide the comprehensive insight into theory of computation by understanding grammar. languages and other elements of modern language design.
- To develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Differentiate between terms used in finite automata.
CO2: Apply various types of automata for solving problems.
CO3: Create and interpret automata models.
CO4: Compare different forms of grammar.
CO5: Construction of PDA and compare it with FA.
CO6: Develop Turing machine and compare with other automata.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Automata Theory and Formal Languages	10
2	Regular sets and regular grammar, PDA and CFG	10
3	Linear bounded automata and Turing machine	10
	Total	**30**
PRACTICALS		-

Module	Theory of Computation	No. of Hours/Credits 30/2
1	**Automata Theory and Formal languages**	**10**
	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 Comsky Classification of Grammar and Languages, Recursive Enumerable Sets,Operations on Languages, Languages and Automata	
2	**Regular sets and regular grammar, PDA and CFG**	**10**
	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	
3	**Linear Bound Automata, Turing Machines**	**10**
	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	

RECOMMENDED READING:

Text Books:

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

Reference Books

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

Program: B. Sc. - Computer Science					**Semester: III**
Course: Java Programming					**Course Code: USMACS302**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2 + 1 =3	25	75

Learning 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.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Students should be able to implement Object oriented programming concepts using basic Java programming elements.

CO2: Students should be able to create Java constructs to handle exceptions and work with input/output, threading and networking in Java.

CO3: Students should be able to use Java Collection Framework and AWT to deal with collections of objects and create GUI respectively.

Outline of Syllabus: (per session plan)

Module	Description	No of hours
1	**Basic elements of Java**	10
2	**Exception Handling, Multithreading, I/O Streams and Networking**	10
3	**Collection Framework, Inner Classes, and AWT**	10
	Total	**30**
PRACTICALS		**30**

Module	Java Programming	No. of Hours/Credits 30/2
1	**Basic elements of Java**	10
	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 String Manipulations: String class, String Buffer class Wrapper Classes: Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes.	
2	**Interfaces, Packages, Exception Handling, Multithreading, File I/O Communication**	10
	Interfaces: Introduction, Implementation & Partial Implementation, Multiple Inheritance Packages : Introduction, Predefined packages, User Defined Packages, Access specifiers 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 File I/O Communication: File reading, File Copying	
3	**File I/O, Collection Framework, Inner Classes, and AWT**	10
	Networking: Introduction, Socket, Server socket, Client – Server Communication Inner Classes: Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class Collection Framework: Introduction, util Package interfaces, List, Set, Map and their classes	

	AWT: Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, CheckBox, Radio Button, Choice, List, Menu, Text Field, Text Area	

PRACTICALS	
Sr. No.	**Topic.**
1	Write programs in Java to implement basic classes.
2	Write programs in Java to implement inheritance
3	Write programs in in Java to implement arrays
4	Write programs in in Java to implement interfaces
5	Write programs in Java to implement packages
6	Write programs in Java to implement threads as by extending the Thread class or implementing Runnable interface
7	Write programs in Java to implement exception handling and create a Java class to create an user-defined exception
8	Write Java Programs to implement file input\output as 1. Read text from a file and print it on screen. 2. Read text from a file and print only numbers into another file.
9	Write Java Programs to implement network communication as 1. Create server-client communication using UDP 2. Create server-client communication using TCP/IP
10	Write Java Programs to use AWT to implement a calculator

RECOMMENDED READING:

Text Books:

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

Reference Books

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/

Program: B.Sc.- Computer Science					**Semester : III**
Course: Operating System					**Course Code: USMACS303**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1=3	25	75

Learning Objectives:

- Learners must understand proper working of operating system.
- To provide a sound understanding of Computer operating system, its structures, functioning and algorithms.

Course Outcomes:

CO1: To interpret the working of operating system, its structures and functioning

CO2: Compare and analyze the scheduling algorithms used by operating systems for various purposes.

CO3: Analyze the working of deadlock

CO4 : Compare the algorithms of disk scheduling algorithm.

CO5:. Analyze the working of page fault and implement the algorithm.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Introduction and Operating-Systems Structures, Processes , Threads	10
2	Process Synchronization, CPU Scheduling, Deadlocks	10
3	Main Memory, Virtual Memory, Mass-Storage Structure, File System	10
	Total	**30**
PRACTICALS		**30**

Module	Operating System	No. of Hours/Credits 30/2
1	**Introduction and Operating-Systems Structures, Processes , Threads**	10
	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 **Threads:** Overview, Multicore Programming, Multithreading Models	
2	**Process Synchronization, CPU Scheduling, Deadlocks**	10
	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	
3	**Main Memory, Virtual Memory, Mass-Storage Structure, File System**	10
	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	

PRACTICALS	
Sr. No.	**Topic.**
1	Implement producer –consumer problem using shared memory.
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.
3	(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.
5	Implement SJF (with no preemption) scheduling algorithm.
6	Implement RR scheduling algorithm.
7	Write a Java program that implements the banker's algorithm
8	Implementation of the FIFO page-replacement algorithm
9	Implementation of LRU page-replacement algorithm
10	Design a File System.

Note : Practical can be performed in any programming language.

RECOMMENDED READING:

Text Books:

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

Reference Books

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

Program: B.Sc. Computer Science				**Semester : III**	
Course: Database Management System				**Course Code: USMACS304**	
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1 = 3	25	75

Learning Objectives:

- To develop capability of concepts and techniques of programming with database.
- Be acquainted with the basics of transaction processing and concurrency control.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Recognize different data types and apply control structure of PL/SQL.

CO2: Discriminate between cursors and devise programs with it.

CO3: Formulate broad range of PL/SQL constructs using stored procedures and triggers.

CO4: Analyze and evaluate various transaction processing, concurrency and recovery control mechanisms.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Fundamentals of PL/SQL, PL/SQL Control Structures, Cursors	10
2	Stored Procedures, Triggers, Sequences	10
3	Transaction Management, DCL Statements, Crash Recovery	10
	Total	**30**
PRACTICALS		**30**

Module	Database Management System	No. of Hours/Credits 30/2
1	**Fundamentals of PL/SQL, PL/SQL Control Structures, Cursors**	**10**
	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. 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 Cursors: Concept of a cursor, types of cursors: implicit cursors; explicit cursor, Cursor for loops, Cursor variables, parameterized cursors	
2	**Stored Procedures, Triggers, Sequences**	**10**
	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	
3	**Transaction Management, DCL Statements, Crash Recovery**	**10**
	Transaction Management: ACID Properties, Serializability, 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	

PRACTICALS	
Sr. No.	**Topic.**

1	Writing PL/SQL Blocks with basic programming constructs by including following: a. Sequential Statements b. unconstrained loop c. If...then...Else, IF...ELSIF...ELSE... END IF
2	Writing PL/SQL Blocks with basic programming constructs by including following: a. Insert value in while loop . CASE WHEN statement with variable c. Use GoTO to jump out of a loop, NULL as a statement inside IF
3	Cursors with a. Types: Implicit Cursor and Explicit Cursor b. Cursor for loops c. A program for simple loop and fetching the cursor. d. Create a cursor in for statement.
4	Cursors with a. A cursor with sub queries b. Combination of PL/SQL, cursor and for loop c. Parameterized cursors d. Cursor Variables
5	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
6	Functions in PL/SQL Block a. Define and call a function b. Define and use function in select clause, Call function in dbms_output.put_line c. Recursive function d. Count Employee from a function and return value back e. Call function and store the return value to a variable
7	Creating and working with Sequences
8	Creating and working with Insert/Update/Delete Trigger using Before/After clause
9	Study of transactions and locks

RECOMMENDED READING:

Text Books:

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.

Reference Books

1. Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education
2. Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner's Guide, Tata McGraw-Hill.
3. George Koch and Kevin Loney ,ORACLE "The Complete Reference", Tata McGraw Hill,New Delhi
4. Peter Rob and Coronel, "Database Systems, Design, Implementation and Management", Thomson Learning

Program: B.Sc.- Computer Science					**Semester : III**
Course: Advanced Web Technologies					**Course Code: USMACS308**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1 = 3	25	75

Learning Objectives:

- To create valid and well-formed XML documents with formatting using XSL
- To incorporate AJAX technology in creating webpages and validate the webpages using server-side scripting
- To create interactive webpages using jQuery, Bootstrap and Joomla

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Design and apply XML to create a markup language
CO2: Incorporate ajax-enabled platform with server-side scripting in website development.
CO3: Develop interactive webpages using jQuery and Bootstrap

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	XML and XSL	10
2	AJAX and PHP	10
3	jQuery, Bootstrap, Joomla	10
	Total	**30**
PRACTICALS		**30**

Module	Advanced Web Technologies	No. of Hours/Credits 30/2
1	**XML and XSL**	**10**
	XML- Introduction to XML, Comparing XML with HTML Describing the Structure of XML - Declaration, Elements, Attributes, Comments, CDATA, XML Entity References, Parsers ,Describing Document Type Definitions, Using XSLT with XML :xsl:template Element , xsl:apply-templates Element, Element,xsl:element Element, xsl:attribute Element, xsl:value-of Element, Using Conditional Statements, Sorting Elements, XSLT functions, Creating Well-formed and Valid Document	
2	**AJAX and PHP**	**10**
	Introduction to Ajax – AJAX Web Application Model, Working of AJAX Asynchronous Data Transfer with XMLHttpRequest-Creating the XMLHttpRequest Object, XMLHttpRequest Properties, XMLHttpRequest Methods, Using the XMLHttpRequest Object in Different Browsers, Reading a File Synchronously, Reading a File Asynchronously, Performing Tasks Using the XMLHttpRequest Object, Integrating PHP and AJAX-Sending Data from a Web Application to a Server, Validating a Field Using AJAX and PHP Handling XML Data using PHP and AJAX-JavaScript properties for Extracting with nodeValue, Accessing XML Elements by Name, Accessing Attribute Values in XML Elements.	
3	**jQuery, Bootstrap, Joomla**	**10**
	Bootstrap (front-end web development framework): Introduction – Responsive Web Design, Bootstrap 4, Adding Bootstrap to website. Layout – Containers, Bootstrap Grid System. Bootstrap with CSS – Typography, Images, Tables. Components – Forms, Buttons, Dropdowns, Jumbotron, Card, List Group, Collapse, Alerts, Navs, Navbar, Progress, Tooltips, Modals, Glyphicons, Carousel. jQuery- JavaScript DOM objects their methods and properties- Window, History, Location Document, Form etc. Fundamentals of jQuery, Loading and using jQuery, using jQuery Library files, Callback functions, jQuery Selectors , jQuery Methods to Access HTML Attributes, jQuery Methods of traversing, jQuery Manipulators, jQuery Events, jQuery Effects, jQuery with AJAX Joomla - Understanding Basic Joomla Template, Customizing Joomla Template, Building Custom Joomla Template, Linking CSS, Linking	

	Javascript, Understanding include Displaying content in HTML,	

PRACTICALS	
Sr. No.	**Topic.**
1	Creating webpage using XML with DTD
2	Formatting XML document with XSL – Basic commands
3	Formatting XML document with XSL – Conditional formatting
4	Basic concepts of AJAX Programming
5	Advanced concepts of AJAX Programming
6	AJAX with PHP
7	AJAX with XML
8	jQuery Practical – Selectors and Manipulators
9	jQuery Practical – Traversal of objects and Events
10	Creating Interactive Webpages using Bootstrap / Joomla

RECOMMENDED READING:

Text Books:

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

Reference Books

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

Program: B.Sc.- Computer Science				**Semester : III**	
Course: Computer Networks				**Course Code: USMACS306**	
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1	25	75

Learning Objectives:

- Demonstrate the concepts of data communication at physical layer and compare ISO - OSI model with TCP/IP model.
- Explore different design issues at all the layers.
- Design the network using IP addressing and sub netting / supernetting schemes.
- Analyze transport layer protocols and congestion control algorithms.
- Explore protocols at application layer.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: utilize concepts and fundamentals of data communication and computer networks.

CO2: explore the inter-working of various layers of OSI.

CO3: explore the issues and challenges of protocols design while delving into TCP/IP protocol suite.

CO4: evaluate various routing algorithms.

CO5: recognize transport layer and application layer protocols.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Introduction to Network Models	10
2	Introduction to Physical Layer and Data-Link Layer	10
3	Network layer & Transport Layer	10
	Total	**30**
PRACTICALS		**30**

Module	Computer Networks	No. of Hours/Credits 30/2
1	**Introduction Network Models:**	10
	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: Layer details of OSI, TCP/IP models, Communication between 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)	
2	**Introduction to Physical Layer and Data-Link Layer:**	10
	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. DLL Design Issues (Services, Framing, Error Control, Flow Control), Error Detection and Correction(Hamming Code, CRC, Checksum) , Elementary Data Link protocols , Stop and Wait, Sliding Window(Go Back N, Selective Repeat)	
3	**Network layer & Transport Layer**	10

	Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA. IPv4 Addressing (classfull and classless), Subnetting, Supernetting design problems, IPv4 Protocol, ARP, Network Address Translation (NAT), IPv6. Transport-Layer Services, Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment. Intra domain & Inter domain Routing, DNS: Name Space, Resource Record and Types of Name Server.	

PRACTICALS	
Sr. No.	**Topic.**
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	Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: *ping, traceroute, netstat, arp, ipconfig.*
4	Using **Packet Tracer** • create a basic network of two computers using appropriate network wire. • create a wireless network of multiple PCs using appropriate access point.
5	Using **Packet Tracer**, connect a network in triangular shape with three layer two switches and every switch will have four computers. Verify their connectivity with each other.
6	Using **Packet Tracer** create a VLAN using GUI.
7	Using **Packet Tracer** create a VLAN using CLI.
8	CRC/ Hamming code implementation.
9	Problem solving with IPv4, which will include concept of Classfull addressing. (supportive Hint: use Cisco Binary Game)
10	Use **Wireshark** to understand the operation of TCP/IP layers : • Ethernet Layer : Frame header, Frame size etc. • Data Link Layer : MAC address, ARP (IP and MAC address binding) • Network Layer : IP Packet (header, fragmentation), ICMP (Query and Echo) • Transport Layer: TCP Ports, TCP handshake segments etc.

	• Application Layer: DHCP, FTP, HTTP header formats

RECOMMENDED READING:

Textbooks:

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.

Reference Books:

1) Computer Network, Bhushan Trivedi, Oxford University Press
2) Data and Computer Communication, William Stallings, PHI

Program: B.Sc.- Computer Science	**Semester : III**
Course: Combinatorics and Graph Theory	**Course Code: USMACS307**

Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1=3	25	75

Learning Objectives:

To give the learner a broad exposure of combinatorial Mathematics through applications especially the Computer Science applications.

Course Outcomes:

CO1 : Apply the combinatorics and combinatorial problems naturally in real life.

CO2: Apply the applications of combinatorial features in real world situations and Computer Science applications.

CO3: Apply the applications of graph theory in real world situations and Computer Science applications.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	**Introduction to Combinatorics**	10
2	**Graph Theory**	10
3	**Network Flows**	10
	Total	**30**
PRACTICALS		**30**

Module	Combinatorics and Graph Theory	No. of Hours/Credits 30/2
1	**Introduction to Combinatorics**	**10**
	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	
2	**Graph Theory**	**10**
	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	
3	**Network Flows**	**10**
	Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm 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	

PRACTICALS	
Sr. No.	**Topic.**
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.

RECOMMENDED READING:

Text Books:

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

Reference Books

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).

Program: B.Sc. - Computer Science					**Semester : IV**
Course: Fundamentals of Algorithms					**Course Code: USMACS401**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1 = 3	25	75

Learning Objectives:

- To understand basic principles of algorithm design and why algorithm analysis is important
- To understand how to transform new problems into algorithmic problems with efficient solutions
- To understand algorithm design techniques for solving different problems

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Examine and evaluate performance of different algorithms

CO2: Derive and solve recurrences describing the performance of divide-and-conquer algorithms

CO3: Analyze and apply string matching algorithm.

CO4: Design optimal solution by applying various algorithm techniques like Dynamic Programming and Greedy Method.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Algorithm Analysis	10
2	Tree, String Matching and Selection Algorithms	10
3	Algorithms Design Techniques	10
	Total	**30**
PRACTICALS		**30**

Module	Fundamental of Algorithms	No. of Hours/Credits 30/2
1	**Algorithm Analysis**	10
	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	
2	**Tree, String Matching and Selection Algorithms**	10
	Generic Trees (N-ary Trees), Threaded Binary Tree, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees, Heapsort String Matching: Introduction, The naive string-matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm 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	
3	**Algorithms Design Techniques**	10
	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, 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 - Longest Common Subsequence, Dynamic Programming Approaches	

PRACTICALS

Sr. No.	Topic.
1	Write Python program to sort n names using Quick sort algorithm. Discuss the complexity of algorithm used.
2	Write Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of algorithm used.
3	Write a Python program to implement Heapsort.
4	Write a python program to implement Rabin-Karp algorithm.
5	Write a python program to implement KMP algorithm.
6	Write Python program for finding the second largest element in an array A of size n using Tournament Method.
7	Write python program to find kth smallest element using partition-based algorithm.
8	Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm.

RECOMMENDED READING:

Text Books:

1. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi , CareerMonk Publications, 2016
2. Introduction to Algorithm, Thomas H Cormen, PHI
3. Design and analysis of algorithms, Himanshu Dave, Pearson, 2nd Edition

Reference Books

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
3. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson

Program: B.Sc.- Computer Science				**Semester : IV**	
Course: Advanced Java				**Course Code: USMACS402**	
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1	25	75

Learning Objectives:

- To Explore swing API for building GUI applications.
- To Understand the Java Database Connectivity.
- To Understand Web application development using Servlet, JSP and sturts2.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Apply swing concepts for creating GUI java applications.

CO2: Implement database connectivity with JDBC.

CO3: Explore servlets, JSP for web development.

CO4: Recognize java beans, struts2 framework and JSON format.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	GUI using Swing & JDBC	10
2	Web Application development with Servlet & JSP	10
3	Java Sturts2 framework & JSON	10
	Total	**30**
PRACTICALS		**30**

Module	Advanced Java	No. of Hours/Credits 30/2
1	**GUI using Swing & JDBC**	10
	Swing: Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: Jlabel, JTextField and JPasswordField, JTextArea, 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.	
2	**Web Application development with Servlet & JSP**	10
	Servlets: Introduction, 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.	
3	**Java Beans, Sturts2 framework & JSON**	10
	Java Beans:-Introduction to JavaBeans-Properties. **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, Data Types, Objects, Schema, Comparison with XML, JSON with Java.	

PRACTICALS	
Sr. No.	**Topic.**
1	Develop the presentation layer of Library Management software application with suitable menus.
2	Develop business logic layer for Library Management System:-1.
3	Develop business logic layer for Library Management System:-2.
4	Write a Java application to demonstrate servlet life cycle.
5	Design a servlet session management application using all the 4 types studied.
6	Implement CRUD operations using servlet(s).
7	Implement CRUD operations using JSP(s).
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	Demonstrate a MVC application using Struts2.

RECOMMENDED READING:

Textbooks:

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

Reference Books:

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

Program: B.Sc.- Computer Science	**Semester : IV**
Course: Physical Computing and IoT Programming	**Course Code: USMACS403**

Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1=3	25	75

Learning Objectives:

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

Course Outcomes:

CO1 : Enable learners to understand System On Chip Architectures.

CO2 : Introduction and preparing Raspberry Pi with hardware and installation.

CO3: Learn physical interfaces and electronics of Raspberry Pi and program them using practical's

CO4 : Learn how to make consumer grade IoT safe and secure with proper use of protocols.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	**Introduction to Microcontroller, Raspberry Pi**	10
2	**M2M to IoT _ A Market, M2M and IoT Technology Fundamentals**	10
3	**IoT and Protocols, IoT Security and Interoperability**	10
	Total	**30**
PRACTICALS		**30**

Module	**Physical Computing and IoT Programming**	**No. of Hours/Credits 30/2**
1	**Introduction to Microcontroller, Embedded system and micro processors**	**10**
	. **Characteristics microcontroller.** **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. **Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Configuring Raspberry Pi with Linux Commands **Programing interfaces:** Introduction to Node.js, **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	
2	**M2M to IoT _ A Market, M2M and IoT Technology Fundamentals**	**10**
	Introduction of M to M : A brief background M2M communication A typical M2M solution overview Key application areas Trends in information and communications technologies **M2M to IoT _ A Market** Perspective Information marketplaces ,Global value chains, M2M value chains, IoT value chains M2M to IoT _ An Architectural Overview Building an architecture **M2M and IoT Technology Fundamentals** Devices and gateways , Local and wide area networking, Data management, M2M and IoT analytics, Knowledge management, **Architecture Reference Model** IoT reference model, Information model, Functional model, Communication model, Safety, privacy, trust, security model	
3	**IoT and Protocols, IoT Security and Interoperability**	**10**
	Introduction to IoT: What is IoT? IoT examples, Simple IoT LED Program. **IoT and Protocols** **IoT Security:** 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.	

PRACTICALS	
Sr. No.	**Topic.**
1	Study of Linux Commands
2	Study of different IC's (8255 , 8259, 8237 and UART)
3	Study and Understanding of Raspberry Pi
4	GPIO: Light the LED with Python
5	Stepper Motor Control: PWM to manage stepper motor speed
6	Study of software used in raspberry pi
7	Case study on IoT in different fields
8	Study / learn any application using raspberry pi
9	GPIO : to control the brightness of LED using Raspberry Pi
10	Stack of Raspberry Pi for better Computing and analysis

RECOMMENDED READING:

Text Books:

1) Learning Internet of Things, Peter Waher, Packt Publishing(2015)
2) Mastering the Raspberry Pi, Warren Gay, Apress(2014)

Reference Books

1. Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly

Program: B.Sc.- Computer Science				**Semester : IV**	
Course: Android Developer Fundamentals				**Course Code: USMACS404**	
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1=3	25	75

Learning 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.

Course Outcomes:

CO1 : Understand the requirements of Mobile programming environment.
CO2 : Learn about basic methods, tools and techniques for developing Apps
CO3 : Explore and practice App development on Android Platform
CO4 : Develop working prototypes of working systems for various uses in daily lives.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	**Android, Activities and Intents**	10
2	**User Input Controls**	10
3	**Data, ContentProviders**	10
	Total	**30**
PRACTICALS		**30**

Module	Android Developer Fundamentals	No. of Hours/Credits 30/2
1	**Android, Activities and Intents**	**10**
	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	
2	**User Input Controls**	**10**
	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	
3	**Data, ContentProviders**	**10**
	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	

PRACTICALS	
Sr. No.	**Topic.**
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

RECOMMENDED READING:

Text Books:

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

Reference Books

1) https://developers.google.com/training/courses/android-fundamentals

2) https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-practicals/details

Program: B. Sc. - Computer Science					**Semester: IV**
Course: Linear Algebra with Python					**Course Code: USMACS405**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2 + 1 =3	25	75

Learning Objectives:

- To offer the learner the relevant linear algebra concepts through computer science applications.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Students should be able to solve linear equations and do various matrix computations

CO2: Students should be able to derive Echelon form, Row canonical form, and deal with basis and change of basis computations

CO3: Students should be able to calculate eigenvalues and diagonalize using them and do various linear transformations.

Outline of Syllabus: (per session plan)

Module	Description	No of hours
1	Vectors, Linear Equations, Matrices	10
2	Solving Systems of Linear Equations, Vector spaces, Vector basis	10
3	Eigenvalues, Eigenvectors and Diagonalization, Inner Product, Orthogonality	10
	Total	**30**
PRACTICALS		**30**

Module	Linear Algebra with Python	No. of Hours/Credits 30/2
1	**Vectors, Linear Equations, Matrices**	**10**
	Vectors: Introduction, Vector addition and multiplication, Dot-product, Cross Product, The geometry of sets of vectors, Complex numbers Matrices: Matrices as vectors, Dot Product, Matrix Multiplication, Transpose, Inverse, Matrix Transformations, Determinant and its properties Linear Equations: Basics, Linear Systems of Equations, Homogeneous Systems and otherwise.	
2	**Solving Systems of Linear Equations, Vector spaces, Vector basis**	**10**
	Solving Systems of Linear Equations: Solving triangular system of linear equations, Gaussian Elimination, Echelon form, Row Canonical form Vector spaces: Vector Spaces, Subspaces, Linear Combinations, Vector Span and Spanning Set, Linear Dependence and Independence Vector Basis: Basis and its Dimension & Rank, Change of Basis, Null space and finding a basis for the null space, The Rank of a Matrix and Applications.	
3	**Eigenvalues, Eigenvectors and Diagonalization, Inner Product, Orthogonality**	**10**
	Eigenvalues, Eigenvectors and diagonalization: Eigenvalues and Eigenvectors, Existence of eigenvalues, Coordinate representation in terms of eigenvectors, Diagonalization Inner Product & Outer product: The inner product for vectors, Outer product for vectors Orthogonalization: Orthogonality, Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors	

PRACTICALS

Sr. No.	Topic.
1	Write a program to perform basic operations of complex number

2	Write a program to perform basic operations of vectors
3	Write a program to perform basic matrix operations
4	Write a program to inverse a matrix
5	Write a program to perform Gaussian Elimination
6	Write a program to convert any matrix to its echelon form
7	Write a program to convert any matrix to its row-canonical form
8	Write a program to enter a given matrix and an eigen value of the same. Find its eigen vector
9	Write a program to perform diagonalization of a matrix given its eigen values and eigen vector.
10	Write a program to do the following: 1. Enter a vector b and find the projection of b orthogonal to a given vector u. 2. Find the projection of b orthogonal to a set of given vectors

RECOMMENDED READING:

Text Books:

1. B. Kolman , D. Hill, Introductory Linear Algebra, An Applied First Course, Pearson Edn; 8th Edn; (2008)
2. Schaum's outlines Linear Algebra, Seymour Lipschutz, Marc Lars Lipson, 4th Edition, McGraw Hill

Reference Books

1. Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012).
2. Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)
3. Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition (2007).
4. Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition
5. H.Anton, Chris Rorres, Linear Algebra with Applns., Wiley, 7th Edn; (1994)

Program: B.Sc.- Computer Science	**Semester : IV**
Course: .NET Technologies	**Course Code: USMACS406**

Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1 = 3	25	75

Learning Objectives:

- To learn to create console applications using C#
- To explore .NET technologies for designing and developing dynamic websites
- To apply database connectivity in .Net Applications
- To create interactive and responsive web applications.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Understand the .NET framework
CO2: Develop a proficiency in the C# programming language
CO3: Proficiently develop ASP.NET web applications using C#
CO4: Incorporate ADO.NET for data persistence in a web application

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	**The .NET Framework, C# Language Basics, ASP.NET**	10
2	**Web Controls, Validation, Master Pages**	10
3	**ADO.NET, Working with XML**	10
	Total	**30**
PRACTICALS		**30**

Module	.NET Technologies	No. of Hours/Credits 30/2
1	**The .NET Framework, C# Language Basics, ASP.NET**	10
	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 -** 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	
2	**Web Controls, Validation, Master Pages**	10
	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, ClientSide Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions **Rich Controls:** Calendar Control, AdRotator Control, MultiView Control ASP.NET Master Pages: 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	
3	**ADO.NET, Working with XML**	10
	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	

PRACTICALS	
Sr. No.	**Topic.**
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	Design ASP.NET page and perform validation using various Validation Controls
6	Design ASP.NET Pages with various Navigation Controls. Design an ASP.NET master web page and use it other (at least 2-3) content pages.
7	Perform ADO.NET data access in ASP.NET for Simple Data Binding
8	Perform ADO.NET data access in ASP.NET for Repeated Value Data Binding
9	Design ASP.NET application for Interacting (Reading / Writing) with XML documents
10	Design ASP.NET Pages for Performance improvement using Caching

RECOMMENDED READING:

Text Books:

1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)

Reference Books

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

Program: B.Sc.- Computer Science					**Semester : IV**
Course: Software Engineering					**Course Code: USMACS407**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	--	-	2	25	75

Learning Objectives:

- To help students to develop skills that will enable them to construct software of high quality
- To make students understand how to develop software that is reliable, and that is reasonably easy to understand, modify and maintain.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Apply the software engineering lifecycle by demonstrating competence in various phases
CO2: Identify the defects and bugs that are present in the software and learn the ways to remove them
CO3: Understand different types of Software Testing techniques
CO4: Work in one or more significant application domains

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Introduction, Project Feasibility Study	10
2	System Analysis, UML	10
3	Software Project Management, Software Measurement and Metrics	10
	Total	**30**
PRACTICALS		--

Module	Software Engineering	No. of Hours/Credits 30/2
1	**Introduction, Project Feasibility Study**	**10**
	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 **Project Feasibility Study** - Operational, Technical, Economic, Organizational and Cultural feasibility. Defining project costs and project benefits. Cost/Benefit Analysis for a project	
2	**System Analysis, UML**	**10**
	Investigating System Requirements – Software Requirement Specification Document, Need of SRS, Characteristics & Components of SRS, Stakeholders, Identifying requirements using various techniques (such as Questionnaires, reviewing reports/forms, interviews, workflows etc) **UML:** Basics of UML, Types of UML Diagrams, Use Case Diagram, Class Diagram, Object Diagram, Sequence diagram & Collaboration diagram, State Transition & State chart diagrams UML Activity Diagram, Component Diagram, Package & Deployment Diagram 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	
3	**Software Project Management, Software Measurement and Metrics**	**10**
	Software Project Management: Estimation in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Project Scheduling - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts **Software Measurement and Metrics:** Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object- Oriented Design, 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 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	

RECOMMENDED READING:

Text Books:

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

Reference Books

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, Springe