Scheme of Instructions & Syllabi

of

Master of Computer Applications

(2025-27)

[As per CBCS guidelines given by UGC]

Total Credit of the Program					
Semester	**I**	**II**	**III**	**IV**	**Total**
Credits	26	24	26	24	**100**

Dr. Akash Sanghi	**Prof. YDS Arya**	**Prof. Manish Gupta**
HOD, Computer Applications	Vice Chancellor	Dean, Computer Applications

Faculty of Computer Applications
INVERTIS UNIVERSITY
Bareilly-243123 U.P.

STUDY AND EVALUATION SCHEME
Master of Computer Applications
(2025-27)

SEMESTER I, YEAR I

SNo	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	MCA101	Advanced Database Management Systems	3+1+0	30	70	100	4
2	MCA102	Advanced Data Structure and Algorithms	3+1+0	30	70	100	4
3	MCA103	Python Programming	3+1+0	30	70	100	4
4	MCA104	Professional Communication-1	3+1+0	25	50	75	3
PRACTICAL / PROJECTS							
5	IIOT13	Introduction to Data Analytics	0+0+4	30	70	100	4
6	MCA151	Advanced Database Management Systems Lab	0+0+4	15	35	50	2
7	MCA152	Advanced Data Structure and Algorithms Lab	0+0+4	15	35	50	2
8	MCA153	Python Programming Lab	0+0+4	15	35	50	2
9	MCA154	Professional Communication-1 Lab	0+0+2	10	15	25	1
TOTAL			**12+4+18**	**200**	**450**	**650**	**26**

STUDY AND EVALUATION SCHEME
Master of Computer Applications
(2025-27)

SEMESTER II, YEAR I

SNo	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	MCA201	Responsive Web Design-Front End Development	3+1+0	30	70	100	4
2	MCA202	Object Oriented Programming Concepts	3+1+0	30	70	100	4
3	MCA203	Advanced Operating Systems	3+1+0	30	70	100	4
4	MCA204	Professional Communication-2	3+1+0	25	50	75	3
PRACTICAL / PROJECTS							
5	IIOT24	Machine Learning	0+0+4	30	70	100	4
6	MCA251	Responsive Web Design-Front End Development Lab	0+0+4	15	35	50	2
7	MCA252	Object Oriented Programming Concepts Lab	0+0+4	15	35	50	2
8	MCA254	Professional Communication-2 Lab	0+0+2	10	15	25	1
TOTAL			**12+4+14**	**185**	**415**	**600**	**24**

STUDY AND EVALUATION SCHEME
Master of Computer Applications
(2025-27)

SEMESTER III, YEAR II

SNo	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	MCA301	MERN Full Stack Development-Backend	3+1+0	30	70	100	4
2	MCA302	Cryptography and Cyber Security	3+1+0	30	70	100	4
3	MCA*	Elective I	3+1+0	30	70	100	4
4	MCA*	Elective II	3+1+0	30	70	100	4
PRACTICAL / PROJECTS							
5	IIOT35	Deep Learning	0+0+4	30	70	100	4
6	MCA351	MERN Full Stack Development-Backend Lab	0+0+4	15	35	50	2
7	MCA*	Elective 1 Lab	0+0+4	15	35	50	2
8	MCA357	Industrial Training Viva **	0+0+0	15	35	50	2
TOTAL			**12+4+12**	**195**	**455**	**650**	**26**

Elective I				**Elective II**		
SNo	**Course Code**	**Course Title**		**SNo**	**Course Code**	**Course Title**
i	MCA313	R–Programming		**i**	MCA318	Advanced Computer Networks
ii	MCA314	Android Programming		**ii**	MCA319	Theory of Computation
iii	MCA315	PHP		**iii**	MCA320	Distributed DBMS
iv	MCA316	GUI using .Net Framework		**iv**	MCA321	Advanced Data Mining Techniques

Elective I Lab		
SNo	**Course Code**	**Course Title**
i.	MCA353	R–Programming Lab
ii	MCA354	Android Programming Lab
iii	MCA355	PHP Lab
iv	MCA356	.Net Framework using C# Lab

STUDY AND EVALUATION SCHEME

Master of Computer Applications

(2025-27)

SEMESTER IV, YEAR II

SNo	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	MCA401	Advanced Java Programming	3+1+0	30	70	100	4
2	MCA*	Elective III	3+1+0	30	70	100	4
PRACTICAL / PROJECTS							
3	IIOT46	Advanced Artificial Intelligence	0+0+4	30	70	100	4
4	MCA451	Advanced Java Programming Lab	0+0+4	15	35	50	2
5	MCA458	Major Project	0+0+20	50	200	250	10
TOTAL			**6+2+28**	**155**	**445**	**600**	**24**

Elective III List					
SNo	**Course Code**	**Course Title**	**S.No**	**Course Code**	**Course Title**
i	MCA412	Design & Analysis of Algorithms	**iv**	MCA415	Big Data Analysis
ii	MCA413	Block Chain Technology	**v**	MCA416	Advanced Soft Computing
iii	MCA414	Cloud Computing & Virtualization	**vi**	MCA417	Data Science

**** Students can choose Electives from Elective Lists.***

*****After 2nd Semester, students will undergo 6 weeks' summer training compulsorily in Public Sector undertakings or Private Sector, known as Industrial Training/Internship. 25 marks will be on viva of students on their Project experience in 3rd Semester.***

Program Outcomes (POs)		
PO1	Computational Knowledge	Understand and apply mathematical foundation, computing and domain knowledge for the conceptualization of computing models from defined problems.
PO2	Problem analysis	Ability to identify, critically analyze and formulate complex computing problems using fundamentals of computer science and application domains.
PO3	Design / Development of Solutions	Ability to transform complex business scenarios and contemporary issues into problems, investigate, understand and propose integrated solutions using emerging technologies
PO4	Conduct Investigations of Complex Computing Problems	Ability to devise and conduct experiments, interpret data and provide well informed conclusions.
PO5	Modern Tool Usage	Ability to select modern computing tools, skills and techniques necessary for innovative software solutions.
PO6	Professional Ethics	Ability to apply and commit professional ethics and cyber regulations in a global economic environment.
PO7	Life-long Learning	Recognize the need for and develop the ability to engage in continuous learning as a Computing professional.
PO8	Project Management and Finance	Ability to understand, management and computing principles with computing knowledge to manage projects in multidisciplinary environments.
PO9	Communication efficacy	Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations.
PO10	Societal & Environmental Concern	Ability to recognize economic, environmental, social, health, legal, ethical issues involved in the use of computer technology and other consequential responsibilities relevant to professional practice.
PO11	Individual & Team Work	Ability to work as a member or leader in diverse teams in multidisciplinary environment.
PO12	Innovation and Entrepreneurship	Identify opportunities, entrepreneurship vision and use of innovative ideas to create value and wealth for the betterment of the individual and society.

MCA101: Advanced Database Management System						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Computer Organization, Operating System, Data Structure, Mathematics

Course Objectives:

1. To introduce the fundamental concepts of database systems, data models, and database architecture.
2. To provide knowledge of relational databases, SQL, and techniques for effective data storage, retrieval, and manipulation.
3. To develop an understanding of database design using Entity-Relationship (ER) modeling and normalization techniques.
4. To enable students to implement and manage databases with attention to data integrity, security, and transaction management.

Detailed Syllabus:

Unit-1 (6 Hours):
Introduction Database Systems: An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Unit-II (10 Hours):
Data Modeling using Relational Data Model: Modeling Techniques- Different Types of Models. Hierarchical Database, Network Database, and Relational Database. Relational data model-Codd's Rules, Concept of Domain, Tuple, and Cardinality. Introduction to ERD-ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation.

Unit-III (10 Hours):
Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependence, lossless join decompositions, normalization using FD, MVD, and JDs.

Unit-IV (10 Hours):
Structured Query Language: Features of SQL, SQL *PLUS, SQL V/s SQL *PLUS, Rules for SQL, SQL Delimiters, Components of SQL. Constraints: Data constraints, Types of data constraints: UNIQUE, NOT NULL at column level, CHECK, NULL value constraint
PL/SQL: Basic Introduction, Advantages of PL/SQL, The generic PL/SQL block, Literals, Variables, Constants, Comparisons, Comments. Control Structure: Conditional Control, Iterative Control and Sequential Control. PL/SQL Transaction: Oracle Transactions, Cursor, Types of Cursor: Implicit cursor, Explicit cursor.

Unit-V (10 Hours):
Transaction Processing Concepts: Transaction system, testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log-based recovery. Concurrency Control- Concurrency control, Protocols for concurrency control-locking,

Time stamping, validation-based protocol. Multiple granularities, multi-versions chemes, Recovery with concurrent transaction.

Unit-VI (10 Hours)

Modern Database Systems: Transaction Processing in Distributed system, data fragmentation, Replication and allocation techniques for distributed system, overview of concurrency control and recovery in distrusted database. Parallel databases, multimedia databases, spatial and temporal databases, data warehousing and data mining, deductive databases.

Text and Reference Books

1. Database System Concepts, Henry Korth , A. Silberschatz, 5th Edition,2005.
2. An Introduction to Database System, Bipin Desai, Galgotia Publications,1991.
3. SQL, PL/SQL the Programming Language of Oracle, Ivan Bayross, BPB Publications, 4thEdition.
4. Schaum's Outline of "Fundamental of Relational Databases", Ramon A. Mata, Pauline K. Cushman, McGraw Hill, December,2006.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts of database systems, data models, and database architecture to design efficient data storage solutions.

CO2: Apply Entity-Relationship (ER) modeling to design relational databases and convert ER diagrams into normalized relational schemas.

CO3: Construct and execute queries using Structured Query Language (SQL) to manipulate and retrieve data effectively.

CO4: Analyze and apply normalization techniques and functional dependencies to optimize database design and reduce data redundancy.

CO5: Understand and implement concepts of transaction management, concurrency control, and database recovery to ensure data integrity and consistency.

MCA102: Advanced Data Structure and Algorithms						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Programming languages, Data Structures, Microprocessor peripherals and interfacing.

Course Objectives:

1. Define and list the functions of an operating system.
2. list resources involved in process creation and management.
3. Explain the use of paging and segmentation
4. Explain the function and structure of the I/O system.
5. Describe path names and directory structure visible to end users

Detailed Syllabus:

UNIT I (10 Hours)

Introduction to Algorithm Design and Data Structures: Abstract data types, Fundamental and derived data types. Representation, Primitive data structures. Algorithm Definition, Analysis of Algorithm, Comparison of Algorithms. Top Down and bottom-up Approaches, Complexity- time and space. Structured approach to programming.

Arrays: Representation of Arrays (Single and Multidimensional arrays), Address calculation using column and row major ordering, Operations on Arrays. Application of arrays- Matrix Multiplication, Sparse matrix.

UNIT II (10 Hours)

Stacks: Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Tower of Hanoi

Queues: Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.

UNIT III (10 Hours)

Linked lists: Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition, Generalized Linked List

UNIT IV (10 Hours)

Trees: Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: In order, Preorder and Post order, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm

UNIT V (6 Hours)

Searching: Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting.

Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees
Hashing: Hash Function, Collision Resolution Strategies Storage Management: Garbage Collection and Compaction.

UNIT VI (10 Hours)
Graphs: Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijkstra Algorithm, Introduction to Activity Networks

Text and Reference Books

1. Data Structures and Program Design in C, R.L. Kruse, B.P. Leung and C. L. Tondo, PHI,2008.
2. Data Structures, Seymour Lipscutz, Mcgraw Hill Publication, 2009
3. Data structures using C, Aaron M.Tenanbaum, Pearson education,2004.
4. Data structure through C, Yashvant Kanetkar, BPB Publication,2006.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts of data structures and their importance in algorithm design and software development.
CO2: Implement linear data structures such as arrays, linked lists, stacks, and queues for various computational problems.
CO3: Apply non-linear data structures like trees and graphs to solve hierarchical and network-based problems.
CO4: Analyze and compare the efficiency of different sorting and searching algorithms in terms of time and space complexity.
CO5: Choose appropriate data structures for problem-solving and develop efficient solutions using structured and modular programming techniques.

MCA103: Python Programming						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Basic computer skills, a problem-solving mindset, and familiarity with programming concepts

Course Objectives:

1. To introduce the fundamentals of Python programming, including syntax, data types, control structures, and functions.
2. To develop the ability to write modular, efficient, and readable code using object-oriented and functional programming concepts in Python.
3. To provide hands-on experience in working with Python libraries for data handling, file operations, and exception handling.
4. To enable students to design and implement real-world applications using Python for problem-solving across domains like web development, data analysis, and automation.

Detailed Syllabus

UNIT I (10 Hours)

GETTING STARTED: History & need of Python, Application of Python, Advantages of Python, Disadvantages of Python, Installing Python, Program structure, Interactive Shell, Executable or script files, User Interface or IDE

PYTHON FUNDAMENTALS: Working with Interactive mode, Working with Script mode, Python Character Set Python Tokens, Keywords, Identifiers, Literals, Operators, Variables and Assignments, Input and Output in Python. DATA HANDLING Data Types Numbers, Strings, Lists, Tuples Dictionary Set, Frozen set, Bool, Mutable and Immutable.

UNIT II (10 Hours)

STRING MANIPULATION: Introduction to Python String, Accessing Individual Elements, String Operators, String Slices, String Functions and Methods.

LIST MANIPULATION: Introduction to Python List, Creating List, Accessing List, Joining List, Replicating List, List Slicing.

TUPLES: Introduction to Tuple, Creating Tuples, Accessing Tuples, Joining Tuples, Replicating Tuples, Tuple Slicing.

DICTIONARIES: Introduction to Dictionary, accessing values in dictionaries, Working with dictionaries Properties.

SET AND FROZENSET: Introduction to Set and Frozen set, Creating Set and Frozen set, Accessing and Joining, Replicating and Slicing. OPERATORS Arithmetic Operators, Relational Operators, Logical Operators, Membership Operators, Identity Operators, Bitwise Operators, Assignment Operators, Operators Precedence, Evaluating Expression, Type Casting.

UNIT III (10 Hours)

PROGRAM CONTROL FLOW: Conditional Statements The if Statement, The if-else Statement, The if-elif Statement, The if-elif Statement, Nested if Statements, Python Indentation, Looping and Iteration, The For Loop, The While Loop, Loop else Statement, Nested Loops, Break and Continue, The Range Function, Introduction to range(), Types of range() function, Use of range() function.

INTRODUCTION TO FUNCTIONS: Built-In Functions, Introduction to Functions, Using a Functions, Python Function Types, Structure of Python Functions, E.g. - map, zip, reduce, filter, any, chr, ord, sorted, globals, locals, all, etc.
USER DEFINED FUNCTIONS: Structure of a Python Program w.r.t. UDF, Types of Functions, Invoking UDF, Flow of Execution, Arguments and Parameters, Default Arguments, Named Arguments, Scope of Variables, Lambda function.
RECURSION FUNCTION: Use of recursion function.

UNIT IV (10 Hours)
MODULES AND PACKAGES: Built-in Modules, Importing Modules in Python Programs, Working with Random Modules, E.g. – built-ins, OS Module, time, datetime.
USER DEFINED MODULES: Structure of Python Modules.
FILE OPERATIONS: Text and Bytes files, opening a file, Reading and Writing Files, Other File tools.

UNIT V (10 Hours)
JSON/PICKLE FORMAT CLASSES AND OBJECTS: Classes as User Defined Data Type, Objects as Instances of Classes, Creating Class and Objects, Creating Objects by Passing Values Variables & Methods.
EXCEPTION HANDLING: Default Exception and Error, Catching Exceptions, raise an exception, try...except statement Raise, Assert, finally blocks.

UNIT VI (10 Hours)
INTRODUCTION TO OOPS: Procedural Vs Modular Programming, The Object-Oriented Programming, Data Abstraction, Data Hiding, Encapsulation, Inheritance, Polymorphism, Generators, Iterators.
DATABASE: Introduction to MySQL, PYMYSQL Connections, Executing queries, Transaction Handling error.
LIBRARY: Introduction to NumPy, Introduction to Pandas and Matplotlib.

Text and Reference Books

1. Python Programming for the Absolute Beginner By Laila M. Dawson
2. Learn Python the Hard Way By Zed A.Shaw
3. Learning Python By Mark Putz Python Documentation (https://docs.python.org)

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the syntax, semantics, and basic constructs of Python programming, including variables, data types, and operators.
CO2: Apply control structures such as decision-making and loops to develop logic-based Python programs.
CO3: Develop modular programs using functions, built-in libraries, and user-defined modules.
CO4: Implement programs using data structures such as strings, lists, tuples, sets, and dictionaries.
CO5: Create file handling programs and apply object-oriented programming concepts such as classes, objects, inheritance, and polymorphism in Python.

MCA104: Professional Communication-1						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of English Grammar and Vocabulary, Fundamental Interpersonal and Presentation Skills

Course Objectives:

1. To enhance students' proficiency in English language skills—listening, speaking, reading, and writing—for professional contexts.
2. To develop effective verbal and non-verbal communication skills required in workplace settings, including interviews, meetings, and presentations.
3. To familiarize students with business and technical communication formats such as reports, emails, proposals, and notices.
4. To cultivate interpersonal skills, confidence, and etiquette necessary for teamwork, leadership, and cross-cultural communication.

Detailed Syllabus

UNIT 1:
FUNDAMENTALS OF COMMUNICATION: Introduction to Communication, Importance of Communication, Elements of a communication process, Types of communication, 7Cs of communication, Barriers to communication, Feedback & Significance, Strategies for effective communication

UNIT 2:
GREETINGS & INTRODUCTION: Importance of greetings, Types of greetings, Etiquette of greetings, Introduction, Components of Introduction, Tips for an effective introduction, Importance of Introduction

UNIT 3:
VOCABULARY EXPANSION & PRONUNCIATION DEVELOPMENT: Importance of Vocabulary expansion, Strategies for Vocabulary expansion, Resources of vocabulary expansion, Pronunciation development & importance, Strategies for pronunciation development, Exercises for pronunciation development

UNIT 4:
PARTS OF SPEECH: Noun, Pronoun, Verb, Adverb, Adjective, Conjunction, Preposition, Articles

UNIT 5:
TENSES & IT'S CLASSIFICATION: Definition, Classification of Tenses: Past, Present, Future

UNIT 6:
INTONATION – RISE & FALL: Understanding Intonation, Functions of Intonation, Types of Intonation patterns, Significance of rising and falling intonation, Practical applications of intonation, Challenges in mastering intonation, Techniques for improving intonation

Text and Reference Books

1. "Technical Communication: Principles and Practice", By Meenakshi Raman & Sangeeta Sharma (Oxford University Press)
2. "Business Communication" By K.K. Sinha, (Galgotia Publishing Company)
3. "Effective Technical Communication" By M. Ashraf Rizvi, (Tata McGraw-Hill Education)
4. "Business Correspondence and Report Writing" By R.C. Sharma & Krishna Mohan (Tata McGraw-Hill Education)

Course Outcomes:

CO1: Demonstrate proficiency in English language skills including listening, speaking, reading, and writing for professional and academic purposes.

CO2: Apply principles of effective verbal and non-verbal communication in interviews, group discussions, and workplace interactions.

CO3: Write clear, concise, and well-structured professional documents such as emails, reports, proposals, and notices.

CO4: Deliver effective presentations and participate confidently in meetings and public speaking situations using appropriate communication tools and techniques.

CO5: Exhibit professionalism, teamwork, and intercultural communication skills essential for career development and workplace success.

MCA201: Responsive Web Design-Front End Development						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: HTML and CSS.

Course Objectives:

1. Understand and Apply the Fundamental Concepts of HTML, CSS, and JavaScript to Build Structured, Responsive, and Interactive Web Pages
2. Evaluate understanding of responsive design principles, advanced CSS techniques, and the use of preprocessors.
3. Design and Implement Responsive Web Interfaces Using Advanced CSS Techniques and Preprocessors to Enhance User Experience Across Devices
4. Write simple JavaScript code to manipulate the DOM and respond to user interactions.
5. Analyze and Implement Advanced JavaScript Techniques to Develop Modular, Efficient, and Maintainable Web Applications.

Module I:

Introduction to HTML:

What is HTML: Definition and purpose of HTML, Role of HTML in web development, Basic syntax and structure

Setting Up Your Development Environment: Text editors (VS Code, Sublime, Notepad++), Browser for testing HTML (Google Chrome, Firefox)

First HTML Page: Creating and saving a basic HTML document, Introduction to <!DOCTYPE html> and basic document structure, Tags: <html>, <head>, <body>

Head Section: The <meta> tag, The <title> tag, The importance of the <head> section

Body Section: Basic text elements: <h1> to <h6>, <p>, <b>, <i>, Paragraphs, line breaks, and horizontal rules, Lists: Ordered (<ol>) and unordered (<ul>) lists, list items (<li>)

Hyperlinks and Images: Creating hyperlinks with <a> tag, Attributes: href, target, title, Adding images with <img> tag, Attributes: src, alt, width, height

Forms and Input Elements: Creating forms: <form> tag, Form elements: <input>, <textarea>, <select>, <button>, Basic form attributes: action, method, name, id

Tables: Table tags: <table>, <tr>, <th>, <td>, Styling tables: border, cellspacing, cellpadding

Multimedia Elements: Embedding audio: <audio>, Embedding video: <video>, Using <source> tag for multiple file formats

Iframes and Embeds: Creating iframes with <iframe>

MODULE II:

HTML5 AND Advanced HTML5 Features :

HTML5 Overview: Introduction to HTML5 features, Key differences between HTML5 and previous versions, Benefits of HTML5 for modern web development

Semantic HTML5 Tags: Introduction to semantic tags: <header>, <footer>, <nav>, <article>, <section>, <aside>, etc.

Audio and Video Elements in HTML5: Using <audio> and <video> tags, Supporting multiple file formats and fallback methods

Forms in HTML5: New input types: email, date, number, range, url, etc., New attributes: required, placeholder, autofocus, pattern, Creating a form with new HTML5 inputs

Local Storage and Session Storage: Introduction to local storage and session storage, Storing and retrieving data with JavaScript, Use cases for local storage in web applications
HTML5 Canvas: Introduction to <canvas> element, drawing basic shapes and graphics with JavaScript, Animation basics using canvas

MODULE-III:
Mastering CSS3:

What is CSS: Definition and purpose of CSS, The role of CSS in web development, How CSS works with HTML to style web pages
CSS Syntax and Structure: CSS rule sets: selectors, properties, and values, Basic syntax: selector {property: value;}, Inline, internal, and external CSS
How to Link CSS to HTML: Inline CSS using the style attribute, Internal CSS within the <style> tag, External CSS with the <link> tag
CSS Selectors: Element selectors (e.g., h1, p), Class selectors (e.g., .class-name), ID selectors (e.g., #id-name), Universal selector (*), Descendant, child, and sibling selectors, Attribute selectors (e.g., [type="text"])
CSS Properties: **Text styling:** color, font-family, font-size, line-height, font-weight, text-align, **Box model:** width, height, margin, padding, border, box-sizing, **Backgrounds:** background-color, background-image, background-position, background-repeat
Understanding the Box Model: Content, padding, border, and margin, Box-sizing property (content-box, border-box), Visualizing the box model using browser developer tools
Layout Techniques: Static vs. relative vs. absolute positioning, floating elements and clearing floats, Centering elements using margin auto, Fixed positioning and sticky positioning
CSS3 Transitions: What are CSS transitions? Transition properties: transition-property, transition-duration, transition-timing-function, transition-delay, Example: Hover effect with transitions
CSS3 Animations: Introduction to keyframes, Creating animations with @keyframes, Animation properties: animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, Practical examples of animations
CSS3 Transforms: transform property: rotate, scale, translate, skew, 2D vs. 3D transforms, Transforming elements on hover (interactive effects)

MODULE-IV:
CSS3 Responsive Design:

Introduction to Responsive Web Design (RWD): What is responsive design? Importance of mobile-first design, Viewport meta tag and its importance in mobile optimization
Media Queries: Syntax of media queries, targeting different devices and screen sizes, Example: Mobile-friendly layout using media queries
Fluid Layouts and Flexible Boxes: Using percentages for fluid widths, Introduction to Flexbox: Basic layout with display: flex, Flexbox properties: justify-content, align-items, flex-wrap, flex-grow, flex-shrink, flex-basis
CSS Grid Layout: Introduction to the CSS Grid system, defining grid containers with display: grid, creating rows and columns with grid-template-rows, grid-template-columns, aligning grid items using justify-items, align-items, and place-items, Example: Building a responsive grid layout
CSS Flexbox and Grid Combined: Combining Flexbox and Grid for complex layouts, Practical use case: Building a multi-column layout with both Flexbox and Grid
Custom Properties (CSS Variables): Introduction to CSS custom properties (variables), Defining and using variables: --primary-color, etc., Benefits of using CSS variables in themes and design consistency
Hover and Focus Effects: Styling links and buttons on hover and focus, Changing background colors, borders, and text styles, Example: Button hover effects with transitions
CSS Shadows and Glows: Box shadows: box-shadow property, Text shadows: text-shadow property, using multiple shadows in one property, Creating glowing effects with shadows
Gradients and Patterns: Linear gradients: background: linear-gradient (), Radial gradients: background: radial-gradient ( ), Repeating gradients and patterns, Example: Background gradient animations

Styling Form Elements: Basic form styles: input fields, buttons, and labels, Input types and custom styles for text, password, email, number, etc., Placeholder and focus effects
Customizing Form Controls: Styling checkboxes, radio buttons, and select dropdowns, Custom form controls with appearance property, Example: Custom form controls with CSS
CSS3 Validation Styles: Styling invalid and valid form inputs, Using the: valid, invalid, required, and: focus pseudo-classes

MODULE-V:
Mastering JavaScript:

What is JavaScript: Definition and role of JavaScript in web development, JavaScript's relationship with HTML and CSS, Running JavaScript: Inline, Internal, and External JavaScript
JavaScript Syntax Basics: Variables: let, const, and var, Basic data types: string, number, Boolean, null, undefined, Simple operators: +, -, *, /, %, ++, --, Comments: single-line and multi-line comments
Conditional Statements: if, else if, else, switch statement
Loops: for loop, while loop, do...while loop, for...of and for...in loops, Loop control: break, continue
Functions: Function declaration and invocation, Parameters and arguments, Return values, Function expressions and arrow functions (() => {}), Scope: Local and Global, Closures and higher-order functions
Arrays: Declaring arrays: let arr = [], Array methods: push (), pop (), shift (), unshift (), slice (), splice (), map (), filter (), reduce (), Accessing and iterating through arrays: for, for Each ()
Objects: Declaring objects: let obj = {key: value}, Accessing and modifying object properties: dot notation and bracket notation, Iterating through objects: for...in loop, Object. Keys (), Object. Values ()
Introduction to the DOM: What is the DOM? (Document Object Model), Accessing elements by ID, class, tag, and query selectors, modifying content and attributes: inner HTML, text Content, set Attribute ()
Event Handling: Adding event listeners: add Event Listener (), Handling different events: click, mouseover, key down, etc., Event propagation: event bubbling and event capturing
ES6+ Features: Let and const vs. var, Arrow functions and this keyword, Template literals (String interpolation): `Hello, ${name}! `, Default parameters in functions, Modules: import and export
Asynchronous JavaScript: Callbacks, promises: then (), catch (), finally (), Async/Await: Simplifying asynchronous code
Error Handling in JavaScript: try...catch block, throwing errors with throw, Custom error messages, Handling asynchronous errors
Debugging JavaScript: Using browser dev tools (Console, Sources, Breakpoints), Common debugging techniques, Debugging asynchronous code

MODULE-VI:
Mastering Bootstrap:

What is Bootstrap: Overview of Bootstrap and its importance, Advantages of using Bootstrap for responsive web design, Installing Bootstrap (via CDN and local setup)
Setting Up Your Development Environment: Text editors: VS Code, Sublime Text, etc., Browsers and developer tools for testing, Linking Bootstrap to your project: CDN vs. downloading
Bootstrap Grid System: Understanding the 12-column grid layout, defining rows and columns with. row and. col-*, Creating responsive layouts using grid breakpoints, Nesting grid columns for advanced layouts
Typography: Bootstrap's default typography styles (headings, paragraphs, lists), Font styles and text alignment: .text-center, .text-left, .text-right, Typography utilities: .font-weight-bold, .text-uppercase, etc.
Buttons: Button classes: .btn, .btn-primary, .btn-success, .btn-danger, Button sizes and block buttons, Button groups and toolbar buttons, Button states: active, disabled, focus, and hover
Images: Responsive images with .img-fluid, Rounded images, circles, and thumbnails with. rounded, rounded-circle, .img-thumbnail, Image alignment and utilities
Icons with Bootstrap: Using Bootstrap Icons or integrating Font Awesome, Applying icons to buttons and links, Icon sizes and alignment

Containers: .container vs. .container-fluid, Fixed-width vs. full-width containers, Responsive containers for different screen sizes
Bootstrap Grid System in Detail: Creating multi-column layouts: col-md-6, col-lg-4, etc., Offsetting and ordering columns, Grid nesting and offsets for complex layouts
Spacing Utilities: Margins and paddings: .m-3, .mt-5,. px-2, Responsive spacing classes, Controlling spacing between elements with margin/padding utilities
Navigation Bar (Navbar): Creating responsive navigation bars using. navbar, Navbar components: brand, links, forms, and dropdowns, Mobile-friendly navbar using the hamburger menu, Customizing navbar background, colors, and alignments
Cards: Using. card for creating card components, Card body, header, footer, and image, Card groups and card decks for multi-card layouts, Styling cards with custom classes
Alerts: Using. alert for displaying messages, customizing alert styles: success, warning, danger, info, Dismissing alerts with JavaScript and. alert-dismissible
Modals: Creating modal dialogs with. modal, Modal header, body, footer, and close button, controlling modal visibility using JavaScript, Using modal for forms, notifications, and more
Forms: Creating forms with. form-control, Input groups for adding icons or buttons to form fields, Custom checkboxes, radio buttons, and selects, Form validation using Bootstrap classes and custom styles
Display and Positioning: Display utilities: d-block, d-inline, d-none, etc., Visibility utilities: visible, invisible, Positioning utilities: position-relative, position-absolute, position-fixed
Flexbox Utilities: Flexbox basics: d-flex, justify-content-*, align-items-*, flex-row, flex-column, Flexbox utilities for centering and alignment, Flexbox ordering and wrapping
Colors and Backgrounds: Text and background color utilities, Color classes: .text-primary, .bg-success, etc., Background utilities for gradients and images
Sizing and Overflow Utilities: Width and height utilities: w-50, .h-100, Controlling overflow: overflow-auto, overflow-hidden
Borders and Shadows: Border utilities: border, border-top, rounded, border-light, Box-shadow utilities: shadow, shadow-lg
Mobile-First Design Philosophy: Why Bootstrap is mobile-first and how it helps responsive design, Understanding responsive breakpoints (xs, sm, md, lg, xl, xxl)
Creating Responsive Layouts: Building a mobile-first, responsive webpage, making images, tables, and forms responsive, Handling mobile navigation with collapsible menus
Carousel: Creating image carousels with. carousel, Carousel controls: next/previous buttons and indicators, Customizing carousel items and controls

INTRODUCTION TO NODE JS
INTRODUCTION TO EXPRESS JS

Project 1 – Responsive Landing Page: Build a simple, responsive landing page with navigation, hero section, and call-to-action button
Project 2 – Blog Layout: Create a responsive blog layout with cards, grid system, and a sidebar navigation
Project 3 – E-commerce Product Page: Design a product page with cards, product details, image gallery, and a modal for product options
Project 4 – Dashboard Layout: Create a responsive admin dashboard with navigation, grid-based layout, cards, and data tables

Text and Reference Books

1. Introduction to frontend technology HTML, CSS and JavaScript Web Publishing language By Laura Lemay, Rafe Colburn, Jennifer kyrnin 1st Edition,2022
2. Theory, practical and critical problem solving in frontend technology (HTML5, CSS3, JavaScript and Adv. JavaScript): Dr. T. Vasudev, Dr. Chandrajit M & Prof. Arvind G.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the core concepts of web development including HTML, CSS, and JavaScript for creating responsive and interactive web pages.
CO2: Develop dynamic web applications using client-server architecture with technologies such as PHP, Node.js, or ASP.NET.
CO3: Design and implement web-based applications with database connectivity using technologies like MySQL or MongoDB.
CO4: Demonstrate the ability to use web development frameworks and tools (e.g., Bootstrap, React, Angular) for efficient web application development.
CO5: Apply security practices, performance optimization, and web hosting techniques in real-world web project deployment.

MCA202: Object Oriented Programming Concepts						
L	**T**	**P**	**Theory**	**Internal**	**Total Marks**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Computer Fundamentals, Principles of computer programming

Course Objectives:

1. To introduce the fundamental principles and features of object-oriented programming including classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
2. To enable students to model real-world problems using object-oriented concepts and design reusable, modular, and maintainable software.
3. To develop the ability to implement object-oriented programs using a suitable programming language such as C++, Java, or Python.
4. To familiarize students with the use of constructors, destructors, operator overloading, exception handling, and file I/O in object-oriented programming.

Detailed Syllabus

UNIT I (10 Hours)

Introduction to OOP: Basic concepts of OOPs, Advantages of OOP, characteristics of object-oriented languages, Object, Classes, Encapsulation, Data Abstraction, Inheritance, Polymorphism, Dynamic binding, Message Passing, keywords, identifiers, data types, manipulators, Operators in C++, Operator Precedence, typecast operator, Control structures, Loops.

UNIT II (6 Hours)

Functions: Function Prototyping, Call by reference, Return by Reference, Default and Constant Arguments, Inline Function, functions Overloading, Friend and virtual Functions, static function.

UNIT III (10 Hours)

Objects and classes: Specifying class & object, Arrays as class member data, Arrays of objects, Constructors and Destructors, objects as function arguments.
Operator Overloading: Overloading Unary & Binary operators,

UNIT IV (10 Hours)

Inheritance: introduction, defining derived classes, overriding member functions, Single Inheritance, multilevel Inheritance, multiple Inheritance, Hierarchical Inheritance, Virtual Base Class. **Files and Streams:** Introduction, classes for file stream operations, opening and closing files, file pointers and their manipulations, Error Handling, command-line Arguments.

UNIT V (10 Hours)

Object Modeling: Objects and classes, links and association, generalization and inheritance, aggregation, abstract class, multiple inheritance, Meta data, candidate keys, constraints. Dynamic **Modeling:** Events and states, operations, nested state diagrams and concurrency, advanced dynamic modeling concepts, a sample dynamic model.

UNIT VI (10 Hours)

Functional Modeling: Data flow diagram, specifying operations, constraints, a sample functional model. OMT, examples and case studies to demonstrate methodologies, comparisons of methodologies, SA/SD, JSD.

Text and Reference Books:

1. Object Oriented Programming with C++, E. Balaguruswamy, 4thEdition.
2. Object Oriented Programming in C++, Robert Lafore, Sams, Dec.,2001.
3. C++ Programming, D. Ravichandran, TMH, 2nd Edition, Dec.2002.
4. Mastering C++, Venugopal, TMH, September,1997.

Course Outcomes:

CO1: Understand and explain key object-oriented concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
CO2: Apply object-oriented techniques to design and implement modular, maintainable, and reusable software systems.
CO3: Develop programs using object-oriented programming languages like C++, Java, or Python to solve real-world problems.
CO4: Demonstrate the use of constructors, destructors, operator overloading, and exception handling in developing robust applications.
CO5: Implement file handling and dynamic memory management techniques using object-oriented principles.

MCA203: Advanced Operating Systems						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	0	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Fundamental Knowledge of Operating Systems, Proficiency in Programming and Data Structures

Course Objectives:

1. To provide in-depth knowledge of the internal structure and functioning of advanced operating systems including distributed, real-time, and multiprocessor OS.
2. To analyze process synchronization, deadlock handling, and inter-process communication mechanisms in complex systems.
3. To explore advanced concepts such as distributed file systems, distributed scheduling, fault tolerance, and security in distributed environments.
4. To enable students to design and implement components of advanced operating systems through case studies and real-time application scenarios.

Detailed syllabus:

UNIT I

Introduction: Definition of operating systems, Computer System architecture: single Processor, Multi-Processor, and Clustered Systems. Operating system structure, Dual Mode Operating system Operations, Distributed System, Operating system services, System calls, system programs, Design Goals, Layered Approach.

UNIT II

Process Management: Process concept, Process scheduling, Cooperating processes, Threads, Inter-process communication, CPU Scheduling: Scheduling Queues, Schedulers, Context Switch, CPU scheduling criteria, Scheduling algorithms, Multiple-processor scheduling.

UNIT III

Process Synchronization and Deadlocks: The Critical-Section problem, Peterson's solution, Semaphores, Classical problems of synchronization, Critical regions, Deadlocks- System model, Characterization: Necessary Conditions, Resource allocation Graph. Deadlock prevention, Avoidance and Detection, Recovery from deadlock.

UNIT IV

Storage management: Memory Management-Basic Hardware, Logical and Physical Address Space, Swapping, Fragmentation, Non-Contiguous Memory allocation, Contiguous Memory allocation, Paging: Basic concept, allocation algorithm, Relocation, Protection. Segmentation: Basic concept, allocation algorithm, Relocation, Protection. Segmentation with paging, Virtual Memory, Demand paging, Page replacement algorithms, Allocation of frames, Thrashing: Cause of Thrashing, Working set Model.

UNIT V

File concept, access methods, and Directory implementation: Linear List, Hash Table. Disk structure, Disk scheduling methods, Disk management: Disk Formatting, Boot Block, Bad Block. Interrupt, Direct Memory Access.

UNIT VI

Security & Case Study: Protection and Security-Goals of protection, Domain of protection, Access matrix, Implementation of access Matrix, The Security problem, Authentication, One Time passwords, Program threats, System threats, Threat Monitoring, Encryption.

Text and Reference Books

1. "Operating system concepts", Galvin, TMH, IV,2006
2. "Operating system concepts & Design", Milan kovic, AddisionWesely,2010.
3. "Operating System", Madnic,TMH, 1997
4. "Operating System", A.s. Godbole, TMH,2001.
5. "Operating System", W.Stallings, Printice Hall, VI, 2007

Course Outcomes:

After completing the course, students will be able to:

CO1: Explain the design principles and architecture of advanced operating systems, including distributed, real-time, and multiprocessor systems.
CO2: Analyze and apply process synchronization, concurrency control, and deadlock handling mechanisms in distributed environments.
CO3: Evaluate and implement inter-process communication techniques and distributed algorithms for effective resource management.
CO4: Design and simulate distributed file systems, security models, and fault-tolerant systems using case studies and real-time scenarios.
CO5: Demonstrate proficiency in using tools and programming constructs to develop and analyze components of advanced

MCA204: Professional Communication-2

L	T	P	Theory	Internal	Total Marks
Hours					
2	1	2	50	25	75

Teaching Scheme	Examination Scheme
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

COURSE OBJECTIVES	
CO 1	Develop inspiring leadership abilities & Master prioritization and time management.
CO 2	Look Sharp, Stay Smart & Etiquette Expert: Practice good manners in all situations.
CO 3	Optimism Overload: Adopt a positive outlook on life & Resilience Rockstar: Bounce back from setbacks.

SYLLABUS

UNIT 1 - EXCELLENCE IN ENGLISH SPEAKING (8 HOURS)

- Importance of proficient English-speaking skills
- Goals and benefits of improving English speaking
- Overview of key components of effective spoken English
- Techniques for improving pronunciation
- Using synonyms, antonyms, and collocations
- Common grammatical mistakes and how to avoid them
- Practicing through speaking drills and exercises
- Strategies for effective communication
- Techniques for engaging in everyday conversations
- Handling small talk and maintaining conversations
- Differences between formal and informal speaking

UNIT 2 - EXPRESSING OPINIONS AND ARGUMENTS RESPECTFULLY (8 HOURS)

- Understanding Opinions and Arguments.
- The importance of listening before responding
- Techniques for expressing agreement clearly and positively
- Techniques for expressing disagreement respectfully
- Role-playing scenarios to practice conflict resolution
- Importance of compromise and negotiation

16th Aug 24

A. Sanghi 20/08/24

22/8/24

UNIT 3 - PROBLEM SOLVING AND CRITICAL THINKING (5 HOURS)

- Problem Solving
- Identifying Problems
- Analysing Problems
- Creative Problem-Solving Techniques
- Implementing Solutions
- Decision Making
- Decision-Making Process
- Evaluating Options
- Making Informed Decisions
- Reflecting on Decisions

UNIT 4 - DIGITAL DETOX (6 HOURS)

- Introduction to Digital Detox
- Identifying signs of digital dependency
- The Impact of Digital Overload
- Assessing Your Digital Usage
- Benefits of Digital Detox
- Mindfulness and Digital Detox
- Planning group activities without digital devices

UNIT 5 - NETWORKING AND BUILDING PROFESSIONAL RELATIONSHIP (8 HOURS)

- Introduction to Networking
- Types of Networking: Formal vs. Informal
- Networking Goals and Objectives
- Building a Professional Network
- LinkedIn: Profile Optimization and Networking Tips
- Other Platforms: Twitter, Facebook, and Industry-Specific Networks
- Managing Your Online Presence on social networks
- Networking Etiquette
- Respecting Boundaries and Privacy
- Networking Challenges and Solutions
- Overcoming Networking Anxiety
- Strategies for Networking in a Remote or Hybrid Work Environment

A. Sanghi 20/08/24

16th Aug 24

22/8/24

LAB

UNIT 1 - SITUATIONAL SPEAKING SKILLS

- Spontaneous Speaking
- Techniques for Thinking on Your Feet
- Practicing Spontaneous Speaking
- Speaking on Various Topics
- Group Discussion
- Participating in a Group Discussion
- Building Consensus
- Handling Conversation in Different Situations

UNIT 2 - PRESENTATION AND PUBLIC SPEAKING

PRESENTATION SKILLS

- Preparing Your Presentation
- Organizing Your Content
- Using Visual Aids
- Practicing Your Presentation

PUBLIC SPEAKING

- Overcoming Stage Fright
- Engaging Your Audience
- Effective Body Language
- Handling Questions and Feedback

UNIT 3 - EMAIL ETIQUETTES AND BUSINESS CORRESPONDENCES

- Introduction to Email Etiquette
- Overview of business correspondence principles
- Structure and Components of a Professional Email
- Clarity and Conciseness
- Professional Language and Grammar in Emails
- Responding to Emails
- Effective Business Correspondence
- Types of business correspondence (e.g., inquiries, requests, confirmations)
- Structuring formal business letters and memos

A. Sanghi 20/08/24

16 th Aug 24

22/8/24

UNIT 4 - CV PREPARATION AND INTERVIEW ETIQUETTES

- Understanding the Purpose of a CV
- The Role of a CV in the Job Application Process
- Structuring Your CV
- Design and Layout
- Preparation for the Interview
- Researching the Company and Role
- Understanding the Interview Format: One-on-One, Panel, or Group
- Preparing Answers for Common Interview Questions
- Professional Appearance
- Conducting Yourself During the Interview
- Demonstrating Enthusiasm and Confidence
- Ethical Considerations and Professionalism

COURSE OUTCOMES	
CO 1	Able to understand fundamentals of business communication strategies.
CO 2	Ability to apply the knowledge of subject practically in real life situations
CO 3	Ability to communicate via electronic mail and other technologies for business messages.
CO 4	Able to develop the proficiency in Language through reading, writing, listening and speaking.
CO 5	Analyze basic principles of critical thinking, problem solving, and technical proficiency in the development of exposition and argument
CO 6	Able to apply business communication strategies and principles to prepare effective communication for domestic and international business.

SUGGESTED READINGS	
Infinity E-book	Improve your communication skills by Alan Barker
English grammar & Composition by Wren & Martin	Science of Effective Communication by Ian Tuhovsky

A Sanghi 20/08/24

16th Aug 24

22/8/24.

MCA301: MERN Full Stack Development- Backend					
L	**T**	**P**	**Theory**	**Internal**	**Total Marks**
Hours					
3	1	4	70	30	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Pre-requisites: DBMS, Manipulate files and installation of software.

Course Objectives:

1. Understand JavaScript fundamentals, including variables, data types, functions, loops, operators, and the Event Loop, to develop basic programming solutions.
2. Apply Advanced JavaScript concepts like arrays, objects, ES6 features, DOM manipulation, asynchronous programming, and API handling using fetch and Axios.
3. Understand the core concepts of Node.js, including its asynchronous, event-driven nature. Learn how to use Nod e.js to build high-performance, non-blocking backend services.
4. Understand the fundamentals of MongoDB, a NoSQL database, and its advantages in modern applications. Learn how to perform basic operations like create, read, update, and delete (CRUD) in MongoDB.
5. Learn Mongoose, an ODM (Object Data Modeling) library that simplifies interaction with MongoDB. Understand how to define schemas, models, and validation for MongoDB documents.

Detailed Syllabus:

Unit-I
Introduction to JavaScript, Features, Scopes, Variables- Var, Let and Const, Data Type, Conditional Statements, Operators, Loops, Comments, Function, Anonymous Function, Map, Filter, Reduce, Event Loop.

Unit-II
Array, Object, Array and Object de-structuring, ES6, Dom Manipulation, Error Handling, Asynchronous JavaScript- Promises, Callbacks, Async Await. Introduction to API – Fetch and AXIOS, JSON – Parse, Stringify

Unit-III
Introduction to React, Introduction to NodeJS, what is node.js, Introduction to NPM, Node Process Model, Setup node.js, Debugging node.js app, Modules in NodeJS, understanding exports and require, Creating modules

Unit-IV
Express JS, Introduction to Express JS, Installation of Express JS, creating server using express JS, Web Browser Building, Express Router, Express Listen Methode, Request matching, Route parameter, Handler function, Request object and Response object, get, post, put, delete Methods, Packages and middleware, Body-parser, cors and credential,

Unit-V

Introduction to MongoDB, Installing MongoDB, the current SQL/NoSQL landscape, Document oriented vs. other types of storage, MongoDB databases, MongoDB Collections, MongoDB Documents, CRUD Operations in MongoDB, MongoDB Methods- insert (), update (), save (), find (), delete (), Working with equality, Query operators, Building complex queries, updating documents, Deleting documents

Unit-VI

Introduction to Mongoose, installing Mongoose, connecting to MongoDB from Mongoose, Core concepts of Mongoose, understanding Mongoose schemas and datatypes, Working with Models, using modifiers in schema, using virtual fields, and Optimizing query performance by enabling indexes.

Text Book:

1. Introduction to Backend Development Technology “Node.js” Design Patterns" by Mario Casciaro
2. "Express.js Guide: The Comprehensive Book on Express.js" by Azat Mardan
3. "Learning MongoDB" by Jason O'Brien, "Mongoose for Application Development" by Simon Holmes

Reference Books:

1. Theory, practical and critical problem solving in backend technology (Node JS, Express JS, Mongoose and MongoDB): by Mr. David Herron, Mr. Douglas Crockford, Mr. Kristina Chodorow

.

Course Outcomes:

At the end of the course, a student will be able to

1. Students will be able to define the programming principles behind theoretical computer science.
2. Students will be able to identify the different computational problems and their associated complexity.
3. Students will understand how to connect client with server, creating a server and HTTP methods.
4. Students will be able to differentiate and give examples for the different types of backend model, Handle file uploads, interactions with cloud services, and external databases.
5. Learn how to use MongoDB to perform advanced data querying, validation, and population (relationships between documents).
6. Integrate the MERN stack (MongoDB, Express.js, React.js, Node.js) and apply deployment strategies to build and host full-stack applications effectively.

MCA302: Cryptography and Cyber Security

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite:
Computer Concepts and C programming, Data Communication & Computer Network

Course Objectives:

1. To introduce the fundamental principles and techniques of cryptography, including symmetric and asymmetric key encryption, hashing, and digital signatures.
2. To understand the key concepts of cyber security, such as authentication, access control, malware threats, and network security protocols.
3. To enable students to analyze and implement secure communication mechanisms and evaluate vulnerabilities in systems and networks.
4. To develop the ability to design and apply cryptographic and cyber security techniques to protect data confidentiality, integrity, and availability in real-world scenarios.

Detailed Syllabus

UNIT I (10 Hours)
Introduction to Cryptography and Network Security: Security Goals, Attacks, Services and Mechanisms, Techniques, Traditional Symmetric Key Cipher.

UNIT II (10 Hours)
Modern Symmetric Key Ciphers: Fiestal Cipher, S-DES, DES, Double DES, Triple DES, AES, Block Cipher. Modes of Operation: ECB, CBC, CFB, OFB and CTR, KDC.

UNIT III (10 Hours)
Introduction to Mathematics for Cryptography: Modular Arithmetic, The Euclidian Algorithm, Extended Euclid, Farmet's and Euler's Theorem, Chinese Remainder Theorem.

UNIT IV (10 Hours)
Asymmetric Key Cryptography: RSA Algorithm, ECC, Key Management- Public Key Distribution, sharing of secret key using A-symmetric Key Cryptosystem.

UNIT V (10 Hours)
Message Authentication: MAC, SHA-512 and MD5. Digital Signature: DSS Key Management: Symmetric Key Distribution, Kerberos.

UNIT VI (10 Hours)
Network Security: IPSec, SSL and TSL, PGP AND S/MIME, SET, System Security: Malicious Software, Firewalls and Intruders.

Text and Reference Books

1. Cryptography and Network Security, Behrouz A Frouzan, TMH, 1st Edition 2007.
2. Cryptography and Network Security, William Stalling, Pearson Education, 4th Edition, 2006.
3. Applied Cryptography, Bruce Schinner, Willy and Sons, 2nd Edition 1996.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand and explain the principles of classical and modern cryptographic techniques, including symmetric and asymmetric encryption.
CO2: Apply cryptographic algorithms such as DES, AES, RSA, and hashing techniques to secure data and communications.
CO3: Analyze various cyber threats and vulnerabilities, and understand the fundamentals of network security, firewalls, and intrusion detection systems.
CO4: Evaluate and implement security mechanisms for authentication, integrity, and confidentiality in real-world applications.
CO5: Demonstrate the ability to use cyber security tools and best practices to design secure systems and respond to cyber-attacks.

MCA313: R-Programming						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite:
Basic Knowledge of Programming Concepts, Basic Understanding of Statistics and Data Handling

Course Objectives:

1. To introduce the fundamental concepts and syntax of R programming for statistical computing and data analysis.
2. To develop skills in data manipulation, cleaning, and visualization using R libraries like dplyr, ggplot2, and tidyr.
3. To enable students to perform basic statistical analysis, hypothesis testing, and exploratory data analysis using R.
4. To prepare students to apply R in real-world data science applications including predictive modeling, data mining, and machine learning.

Detailed Syllabus

UNIT I:
GETTING STARTED WITH R: Installing R - The R environment - R packages - Basics of R - Data Structures - Reading data into R - Graphics in R.

UNIT II:
FUNCTIONS AND STATEMENTS: Writing R functions - Control Statements (if and else, switch, if else, compound tests) - Loops in R (for, while, controlling loops) - Applications using the functions and loops.

UNIT III:
DATA MANIPULATION AND ANALYSIS: Group manipulation - Data Reshaping - Manipulating Strings - Basic Statistics using R (Summaries, Correlation, t-tests, ANOVA)

UNIT IV:
LINEAR MODELS USING R: Linear Models - Simple and Multiple regression, GLM - Logit Regression, Model
diagnostics - Residuals, Cross validation, Boot strapping.

UNIT V:
NON-LINEAR MODELS, TIME SERIES AND CLUSTERING USING R: Nonlinear Models - Non-Linear least square, Splines, Generalized Additive Models, Decision trees, Random forests.

UNIT VI:
TIME SERIES: Time Series - Autoregressive moving average, VAR, GARCH. Clustering - K means, PAM and Hierarchical Clustering.

Text and Reference Books

1. "The Art of R Programming" by Norman Matloff, No Starch Press
2. "Hands-On Programming with R" by Garrett Grolemund, O'Reilly Media
3. "R for Data Science" by Hadley Wickham & Garrett Grolemund, O'Reilly Media
4. "R Programming for Data Analytics", Roger D. Peng, Leanpub

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamentals of R syntax, data types, and control structures for effective programming.
CO2: Perform data manipulation and transformation using R packages such as dplyr, tidyr, and readr.
CO3: Apply statistical techniques and functions in R to analyze and interpret datasets.
CO4: Create data visualizations using R's graphical capabilities including base graphics and ggplot2.
CO5: Develop R scripts for real-world data analysis tasks and apply machine learning libraries for predictive modeling.

MCA314: Android Programming

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	4	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Basics of Java language and PL/SQL

Course Objectives:

1. To introduce students to the Android platform, architecture, and development environment using Android Studio.
2. To develop the ability to design and build interactive and user-friendly mobile applications using Java/Kotlin and XML.
3. To enable students to utilize Android components such as Activities, Intents, Services, Broadcast Receivers, and Content Providers.
4. To provide hands-on experience in integrating mobile apps with databases (e.g., SQLite) and web services (e.g., REST APIs).

Detailed Syllabus

UNIT I

JAVA Concepts: Platform Independency, OOPs Concepts, Inheritance in detail, Exception handling, Packages & interfaces, JVM & .jar file extension, Multi-threading (Thread class & Runnable Interface).SQL: DML & DDL Queries in brief.

UNIT II

Introduction to Android: Introduction of Android, setting up development environment, Installing the SDK, Creating Android Emulator, Android development Tool. Fundamentals: Basic Building blocks - Activities, Services, Broadcast Receivers & Content provider, UI Components - Views & notifications, Components for communication -Intents & Intent Filters, Android API levels (versions & version names)

UNIT III

Application Structure: AndroidManifest.xml, uses-permission & uses-sdk, Resources & R.java, Assets, Layouts & Draw-able Resources, Activities and Activity lifecycle, First sample Application.

UNIT IV

Emulator-Android Virtual Device: Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS. Second App: (switching between activities), Develop an app for demonstrating the communication between Intents.

UNIT V

Basic UI design: Form widgets, Text Fields, Layouts, [dip, dp, sip, sp] versus px, Examples
Preferences: Shared Preferences, Preferences from xml, Examples.

UNIT VI

Menu: Option menu, Context menu, Sub menu, Menu from xml, Menu via code, Examples
UI design: Time and Date, Images and media, Composite, Alert Dialogs & Toast, Popup, Examples

Text and Reference Books

1. Android Application Development (With Kitkat Support), Black Book, by Kogent Learning Solutions Inc. by Pradeep Kothari
2. Android Application Development Cookbook: 93 Recipes for Building Winning Apps (WROX),by Wei-Meng Lee
3. Professional Android 4 Application Development, by Reto Meier
4. Beginning Android 4 Application Development, Wei-Meng Lee
5. Android Application Development, by Lombardo John and Blake Meike

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamentals of Android architecture, lifecycle, and development environment.
CO2: Design and develop user interfaces using Android UI components and layouts.
CO3: Implement core Android components such as Activities, Services, Broadcast Receivers, and Content Providers.
CO4: Develop Android applications that interact with internal (SQLite) and external databases or APIs.
CO5: Build and deploy functional Android apps with features like data storage, multimedia, and location services.

MCA315: PHP						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Course Objectives:

1. To introduce the fundamentals of server-side scripting using PHP for dynamic web development.
2. To enable students to handle form data, manage sessions, and interact with databases using PHP and MySQL.
3. To develop problem-solving skills through the creation of interactive and data-driven web applications.
4. To familiarize students with file handling, error handling, and security practices in PHP programming.

Detailed Syllabus

Unit-1

Introduction to PHP: Evaluation of PHP, Basic Syntax, defining variable and constant, PHP Data type, Operator and Expression, Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with Html.

Unit-2

Function: What is a function, define a function, Call by value and Call by reference, Recursive function, PHP GET and POST, Built-in Functions, User-Defined Functions, Functions with Parameters, Values and arguments in Function.

Unit-3

String and Array: - String - Creating and accessing String, Searching & Replacing String, Formatting String, String Related Library function, Array- Anatomy of an Array, creating index based and Associative array, accessing array Element, Looping with Index based array, looping with associative array using each () and foreach (), Some useful Library function

Unit-4

Introduction to OOPS- Introduction, Objects, declaring a class, the new keyword and constructor, Destructor, Access method and properties using $this variable, Public, private, protected properties and methods, Static properties and method, Class constant, Inheritance & code reusability, Polymorphism, Parent: & self: keyword, Instance of operator, Abstract method and class, Interface, Final

Unit-5

Exception Handling, file and Directories: - Understanding Exception and error, Try, catch, throw, Global Exception Handler, Defining Custom Exceptions, understanding file& directory, Opening and closing a file, Coping, renaming and deleting a file, working with directories.

Unit-6

Database Connectivity with MySql:- Introduction to RDBMS, Connection with MySql Database, performing basic database operation (DML) (Insert, Delete, Update, Select), Executing query.

Text and Reference Books

1. Lynn Beighley & Michael Morrison- Head First Php & MySQL.
2. Robin Nixon: Learning Php, MySQL, Java script and CSS: A step-by-step guide to creating dynamic websites.
3. Luke Welling & Laura Thompson: PHP & MYSQL web development

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the basic syntax, variables, data types, and control structures of PHP.
CO2: Create dynamic web pages using PHP with HTML forms and user input handling.
CO3: Implement session management and cookie handling for stateful web applications.
CO4: Connect PHP applications with MySQL databases to perform CRUD operations.
CO5: Develop secure and efficient web applications with file handling, error handling, and form validation in PHP.

MCA316: GUI using .Net Framework

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	4	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: OOPs Concepts, GUI Interfaces, HTML and CSS.

Course Objectives:

1. To introduce the fundamentals of the .NET framework and its architecture for developing GUI-based applications.
2. To enable students to design and implement graphical user interfaces (GUIs) using Windows Forms and relevant .NET controls.
3. To provide knowledge of event-driven programming and how user interactions are handled in a .NET environment.
4. To develop practical skills in building, debugging, and deploying GUI-based desktop applications.

Detailed Syllabus:

Unit-1

The .Net framework: Introduction of .Net, The Origin of .Net Technology (OLE technologies, COM technologies, .NET technologies), The architecture of .Net Framework, Common Language Runtime (CLR), Common Type System (CTS), Common Language Specification (CLS), Microsoft Intermediate Language (MSIL), Just-In –Time Compilation, Framework Base Classes.

Unit-2

Introduction of Programming Language C#: Introduction of C#, Characteristics of C#, Differences between C# and C++, Differences between C# and JAVA, C# program introduction: The Main method specification, Namespace, Variables: Declaring implicit and explicit variables, Data-types, Boxing and Un-boxing.

Unit-3

Controlling program execution: IF statements, CASE (switch) statements, Operators, Looping, Storing multiple values with arrays. Inheritance, Method Overloading and method overriding, Polymorphism, Operator Overloading, Abstract Class, Inner Class, Interface, Delegates, Partial Classes, Errors and its types, Exception Handling.

Unit-4

GUI –Controls and Their Event Handling: Text Box, Rich Text Box, Masked Text-box, Label, Link Label, Radio Button, Check Box, List Box, Combo Box, Checked List Box. Date Time Picker Control, Calendar Control, Tool Tip, Shock Web Flash Object.

Navigation Control and Its Event Handling: Context Menu Strip, Tool Strip, Status Strip, Tool Strip Container.

Unit-5

Containers and its Event Handling: Flow Layout Panel, Group Box, Panel, Split Container, Tab Control, Table Layout Panel.

Dialog Boxes and its Event Handling: Message Dialog Boxes, Color Dialog, Folder Browser Dialog, Font Dialog, Open File Dialog, Save File Dialog.

Unit-6

Data Controls: Data Source, Data Set, and Data Grid View displaying Record in the Grid View Controls. ADO.Net: Connected and Disconnected Architecture, Displaying Record from the Database, Inserting Record into Database, Creating Login using Database, Deleting Record from the Database, Fetching Record from the Database, Update Record in the Database, Creating Setup of .Net Application using Set up Wizard.

Text and Reference Books

1. Beginning Visual C# 2008, John Wiley, Wrox, May 2008.
2. Microsoft .Net for Programmers, Fergal Grimes, SPI, 2002.
3. Programming with C#, E. Balagurusamy, TMH, 1st Edition.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the architecture and components of the .NET Framework for developing GUI-based applications.

CO2: Design and implement user-friendly graphical interfaces using Windows Forms and standard .NET controls.

CO3: Apply event-driven programming techniques to handle user interactions and system events effectively.

CO4: Develop desktop applications using C# or VB.NET with features such as menus, dialogs, and data input validation.

CO5: Debug, test, and deploy GUI applications while ensuring usability, responsiveness, and maintainability.

MCA318: Advanced Computer Networks						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	0	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Familiarity with the fundamentals of Digital Electronics, A network simulation method.

Course Objectives:

1. Learn how computer network hardware and software operate.
2. Investigate the fundamental issues driving network design.
3. Learn about dominant network technologies.

Detailed Syllabus

Unit-1

Data Communications: Components – Direction of Data flow – Networks – Components and Categories – Types of Connections – Topologies –Protocols and Standards – ISO / OSI model, Example Networks such as ATM, Frame Relay, ISDN Physical layer: Transmission modes, Multiplexing, Transmission Media, Switching, Circuit Switched Networks, Datagram Networks, and Virtual Circuit Networks.

Unit-2

Physical Layer: Concept of Analog & Digital Signal, Bandwidth, Transmission Impairments: Attenuation, Distortion, Noise, Introduction to Transmission Media: Twisted pair, Coaxial cable, Fiber optics, Wireless transmission (radio, microwave, infrared), Switching methods, integrated services digital networks

Unit-3

Medium Access sub layer: Channel Allocations, LAN protocols -ALOHA protocols, Collision free Protocols-Token Passing, IEEE standards, Ethernet and Token Ring. Data Link Layer: Framing, Error detection and correction codes: checksum, CRC, hamming code, Sliding Window Protocols: Stop & Wait ARQ, Go-back-N ARQ, Selective repeat ARQ

Unit-4

Network Layer: Point-to Point networks, Routing algorithms, Congestion control algorithms, Internetworking Devices, IP protocol, IP addresses: IPv4 classful and classless addressing, Introduction to IPv6

Unit-5

Transport Layer: Connection management: Three-way Handshaking. Introduction of User Datagram Protocol (UDP), Basics of Transmission Control Protocol. (TCP).

Unit-6

Application Layer: File Transfer Protocol, Domain Name System, Electronic mail, Intro of Client server model, Hyper Text Transfer Protocol, WWW, Example Networks - Internet and Public Networks

Text and Reference Books

1. Database System Concepts, Henry Korth , A. Silberschatz, 5th Edition, 2005.
2. An Introduction to Database System, Bipin Desai, Galgotia Publications, 1991.
3. SQL, PL/SQL the Programming Language of Oracle, Ivan Bayross, BPB Publications, 4th Edition.
4. Schaum's Outline of "Fundamental of Relational Databases", Ramon A. Mata, Pauline K. Cushman, McGraw Hill, December, 2006.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the basic concepts of data communication, network models, and transmission media used in modern communication systems.
CO2: Explain the functions and protocols of different layers in the OSI and TCP/IP network models.
CO3: Analyze error detection and correction techniques, as well as flow and congestion control mechanisms in reliable data transfer.
CO4: Evaluate different switching techniques, IP addressing schemes, and routing algorithms used in network communication.
CO5: Demonstrate knowledge of various network topologies, protocols, and security measures used in wired and wireless networks

MCA319: Theory of Computation

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Basic Knowledge of Discrete Mathematics, Fundamentals of Programming and Algorithms

Course Objectives:

1. To introduce the fundamental concepts of formal languages, grammars, and automata that form the theoretical foundation of computer science.
2. To develop the ability to design and analyze finite automata, pushdown automata, and Turing machines for solving computational problems.
3. To provide an understanding of the classification of languages and computational models in terms of their expressive power and limitations.
4. To enable students to explore decidability, reducibility, and the concept of computational complexity for determining problem solvability.

Detailed Syllabus:

UNIT I

Introduction: Basic Concepts: Formal proofs, Additional form of Proofs, Inductive proof, Sets, Relation, Kleen Closures, Graphs, Trees, Symbol, Alphabets, strings and languages, automata and grammar, Applications of automata theory.

UNIT II

Finite Automata: Basic Machine and Finite State Machine. Finite Automata: Definition and Types of Automata- DFA, NFA, Construction of DFA and NFA, NFA with epsilon-Moves, Minimization Of FA, Equivalence of NFA and DFA, Conversion of NFA with epsilon moves to NFA, Conversion of NFA with epsilon moves to DFA, Moore and Mealy Machines, Inter-conversion between Moore and Mealy Machines.

UNIT III

Regular Expressions, Regular Grammar and Languages: Definition and Identities of Regular Expressions, regular and non-regular language, operations on RE and their precedence, Algebraic laws for RE, Regular Expression and Finite Automata, Conversion from RE to FA and DFA to RE, Arden's theorem, Pumping Lemma for RL.

UNIT IV

Context Free Grammar and Languages: Definition and Construction of CFG, Definition and Generation of CFL from CFG, Derivation, derivation trees, Ambiguous Grammar and Removal of Ambiguity. Simplification of CFGs. Normal Forms of Grammar: CNF and GNF.

UNIT V

Pushdown Automata: Definition of push down automata, The language of PDA, Definition and Construction of DPDA and NPDA. Equivalence of PDAs and CFGs, Closure Properties Of CFLs.

UNIT – VI

Turing Machines: Definition and Construction of Turing Machines. Languages of TM. Types of TM. Comparison And Applications of DFA, PDA and TM.

Text and Reference Books

1. John C. martin, "Introduction to Language and Theory of Computation", TMH, Third Edition.
2. Michel Sipser "Introduction to Theory of Computation" Thomson Course Technology, Second Edition
3. Kavi Mahesh, "Theory of Computation" Wiley-India.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand and describe the foundational concepts of formal languages, grammars, and automata theory.
CO2: Design and analyze deterministic and non-deterministic finite automata for recognizing regular languages.
CO3: Construct context-free grammars and pushdown automata to represent context-free languages and solve parsing problems.
CO4: Demonstrate the working of Turing machines and analyze their capability to model any computation.

MCA320: Distributed DBMS

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Database management system

Course Objectives:

1. To introduce the fundamental concepts, architecture, and design principles of distributed database systems.
2. To develop understanding of data distribution strategies, fragmentation, replication, and transparency issues in DDBMS.
3. To enable students to design and implement distributed query processing, transaction management, and concurrency control mechanisms.
4. To expose students to real-world challenges like fault tolerance, data consistency, and distributed recovery in multi-site environments.

Detailed Syllabus:

UNIT I

Introduction: Distributed Data processing, Distributed Database Systems (DDBMSs), Promises of DDBMSs, Complicating factors and Problem areas in DDBMSs, Overview Of Relational DBMS, Relational Database concepts, Normalization, Integrity rules, Relational Data Languages, Relational DBMS.

UNIT II

Distributed DBMS Architecture: DBMS Standardization, Architectural models for Distributed DBMS, Distributed DBMS Architecture. Distributed Database Design: Alternative design Strategies, Distribution design issues, Fragmentation, Allocation. Semantic Data Control: View Management, Data security, Semantic Integrity Control.

UNIT III

Overview of Query Processing: Query processing problem, Objectives of Query Processing, Complexity of Relational Algebra operations, characterization of Query processors, Layers of Query Processing.

UNIT IV

Introduction to Transaction Management: Definition of Transaction, Properties of transaction, types of transaction. Distributed Concurrency Control: Serializability theory, Taxonomy of concurrency control mechanisms, locking based concurrency control algorithms.

UNIT V

Parallel Database Systems: Database servers, Parallel architecture, Parallel DBMS techniques, Parallel execution problems, Parallel execution for hierarchical architecture. Database Interoperability: Database Integration, Query processing.

UNIT VI

Distributed Object Database Management systems: Fundamental Object concepts and Object models, Object distribution design. Architectural issues, Object management, Distributed object storage, Object query processing. Transaction management.

Text and Reference Books

1. Principles of Distributed Database Systems, M. Tamer Ozsu, Patrick Valduriez, 2nd Edition, 1999.
2. Distributed Databases principles and systems, Stefano Ceri, Giuseppe Pelagatti, TMH, 2008.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts and architecture of distributed database systems.
CO2: Apply data fragmentation, allocation, and replication techniques in distributed environments.
CO3: Analyze and implement distributed query processing and optimization strategies.
CO4: Demonstrate concurrency control, transaction management, and recovery techniques in distributed systems.
CO5: Evaluate and design distributed databases for real-time applications with considerations for consistency, availability, and fault tolerance.

MCA321: Advanced Data Mining Techniques						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	0	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Familiarity with the data base management system, Knowledge of repository system.

Course Objectives:

1. To provide in-depth knowledge of advanced data mining algorithms and techniques such as clustering, classification, association, and anomaly detection.
2. To enable students to apply data preprocessing, feature selection, and dimensionality reduction techniques on large datasets.
3. To develop analytical and practical skills to implement data mining methods using tools like R, Python, or Weka for real-world problem-solving.
4. To prepare students for research and application of data mining in specialized domains like web mining, text mining, and big data analytics.

Detailed Syllabus

UNIT I (10 Hours)
Introduction: Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Major issues in Data Mining. Data Pre-processing: Needs, Pre-processing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

UNIT II (10 Hours)
Introduction: Data Warehouse and OLAP Technology for Data Mining, Data Warehouse Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, From Data Warehousing to Data Mining. Data Mining Primitives, Data Mining Query Languages.

UNIT III (10 Hours)
Concepts Description: Characterization and Comparison, Data Generalization and Summarization-Based Characterization. Analytical Characterization, Analysis of Attribute Relevance, Mining Class Comparisons: Discriminating between Different Classes, Mining Descriptive Statistical Measures in Large Databases.

UNIT IV (10 Hours)
Mining Association Rules in Databases: Association Rule Mining, Mining Single-Dimensional Boolean Association Rules from Transactional Databases, Mining Multilevel Association Rules from Transaction Databases, Mining Multidimensional Association Rules from Relational Databases and Data Warehouses, From Association Mining to Correlation Analysis.

UNIT V (6 Hours)
Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Back-propagation, Classification Based on Association Rule Mining, Other Classification Methods, Prediction, and Classifier Accuracy.

UNIT VI (10 Hours)

Cluster Analysis Introduction: Types of Data in Cluster Analysis, a Categorization of Major Clustering Methods, Partitioning Methods, Density-Based Methods, Outlier Analysis. Mining Complex Types of Data: Multidimensional Analysis and Descriptive Mining of Complex, Data Objects, Mining-Spatial Databases, Multimedia Databases, Time-Series and Sequence Data, Text Databases, World Wide Web.

Text and Reference Books

1. Data Mining -Concepts and Techniques, Han, Kamber, Harcourt India, 2006.
2. Data Mining Introductory and advanced topics, Margaret H Dunham, Pearson, 2002.
3. Data Mining Techniques, Arjun K. Pujari, University Press, 2001.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand and explain advanced data mining concepts, models, and algorithms including classification, clustering, and association rule mining.
CO2: Apply preprocessing and feature selection techniques to prepare complex datasets for mining.
CO3: Implement advanced data mining techniques using tools like Python, R, or Weka on real-world datasets.
CO4: Analyze the performance and suitability of different mining algorithms for specific data-driven applications.
CO5: Demonstrate the application of data mining in emerging areas such as web mining, text mining, and big data analytics.

MCA401: Advanced Java Programming						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: C Programming, and OOPs Concepts.

Course Objectives:

1. To provide a strong foundation in Java programming beyond the basics, focusing on advanced features such as JDBC, Servlets, JSP, and networking.
2. To enable students to develop dynamic, interactive, and database-driven web applications using the Java EE platform.
3. To impart knowledge on multithreading, event handling, and network programming to build scalable Java applications.
4. To develop the ability to integrate client-server architecture and web technologies using Java frameworks and APIs.

Detailed Syllabus:

UNIT–1:

Introduction of Java: Features of Java Language, Platform Independency, JVM, Byte-code, Operator, Data type, Variables, Robustness. OOPS: Object, Class, Classifications, Methods & classes, Inheritance, Static and non Static methods.

UNIT–2:

Advanced OOPS Concepts: Call by Value, Call by Reference, Method Overloading, Method Overriding, Abstraction, Interface, Polymorphism, Inner Class & Anonymous Classes, Abstract Class.

UNIT–3:

Conditional Construct in Java, Array & Packages: if, if else, nested if else, if else ladder, Ternary Operator, Switch. Array: Introduction of arrays, Understanding and working with single, double dimensional arrays, Initialization of array, Linear and Binary Search. Packages: Data Encapsulation, Concept of Package, creating package, Importing packages, Child Packages.

UNIT–4:

JDBC: Introduction to JDBC, What is JDBC, Why JDBC is required, JDBC Architecture, JDBC Components, JDBC Driver, Type 1: JDBC-ODBC Bridge, Type 2: Native API Driver, Type 3: Network Protocol Driver, Type 4: Thin Driver, JDBC API, java.sql package, Important Interfaces, Driver, Connection, Statement, PreparedStatement, CallableStatement, ResultSet, JDBC Connectivity Steps, Load Driver, Establish Connection, Create Statement, Execute Query, Process Result, Close Connection, Database Operations, Insert Records, Update Records, Delete Records, Select Records, PreparedStatement, Advantages over Statement, Parameterized Queries, ResultSet, Forward Only, Scrollable ResultSet, Transaction Management, AutoCommit, Commit & Rollback.

UNIT–5:

Java Servlets: Servlet Basics, Servlet Introduction, Servlet Architecture, Servlet API (javax.servlet, javax.servlet.http), Servlet Life Cycle, init(), service(), destroy(), Request & Response Handling, HttpServletRequest, HttpServletResponse, GET vs POST, Database Connectivity using Servlet, Connecting Servlet with Database using JDBC, Using JDBC inside Servlet, CRUD Operations using Servlet, Servlet with JDBC, Login Application (DB-based Authentication), Registration Form (Insert data into DB), Display Records from Database, Session Tracking, Cookies, HttpSession, Request Dispatching, RequestDispatcher, sendRedirect().

UNIT–6: JSP:
JSP Basics, JSP Introduction, JSP vs Servlet, JSP Life Cycle, JSP Elements, Scriptlet, Expression, Declaration, JSP Directives, page, include, JSP Implicit Objects, request, response, session, application, out, JDBC using JSP, JDBC Code inside JSP (basic understanding), Connecting JSP with Database, Displaying Database Records in JSP, JSP with JDBC, Login Page using JSP & JDBC, Registration Page using JSP & JDBC, Display Data in Table Format.

Suggested Readings:

1. The Complete Reference: Java, Herbert Schieldt, TMH, 7th Edition 2006
2. Programming in JAVA, E. Balagurusamy, TMH, 2nd Edition 2007
3. Object Oriented Modeling and Design, James Rumbaugh et al, PHI, 4th Edition 2003
4. Object Oriented Analysis & Design with Application, Booch Grady, Pearson Education, New Delhi, 3rd Edition, 2006.

Course Outcomes:
After completing the course, students will be able to:

CO1: Understand and implement advanced Java concepts such as JDBC, Servlets, and JSP for dynamic web development.
CO2: Develop database-driven applications using Java Database Connectivity (JDBC) with backend integration.
CO3: Design and deploy server-side programs using Servlets and JavaServer Pages (JSP).
CO4: Apply concepts of multithreading, socket programming, and network communication in Java-based projects.
CO5: Create real-world enterprise-level applications by integrating client-server technologies and MVC architecture.

MCA412: Design and Analysis of Algorithm						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	0	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - C Programming Concepts, Data Structure Concepts, Discrete Mathematics concepts.

Course Objectives:

1. To introduce the fundamental concepts of algorithm design techniques such as divide and conquer, greedy algorithms, dynamic programming, and backtracking.
2. To develop the ability to analyze the time and space complexity of algorithms using asymptotic notations.
3. To enable students to design efficient algorithms for solving computational problems and to compare various algorithmic strategies.
4. To familiarize students with advanced topics like NP-completeness, approximation algorithms, and optimization strategies in algorithm design

UNIT I (10 Hours)

Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis of algorithm- Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, Recurrences and their solutions, Amortized analysis.

Divide and Conquer: General method, applications-Binary search, Quick sort, Merge sort, Heap Sort, Strassen's matrix multiplication.

UNIT II (9 Hours)

Advanced Data Structure: Red Black Tree, Binomial Heap, B tree, Fibonacci Heap.

Disjoint Sets: disjoint set operations, union and find algorithms, spanning trees, connected components and bi-connected components.

UNIT III (10 Hours)

Greedy method: General method, Applications-Job sequencing with deadlines, 0/1 knapsack problem, Minimum cost spanning trees.

Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, Travelling sales person problem.

UNIT II (9 Hours)

Graph Algorithm: Graph Algorithms, BFS, DFS, Minimum Spanning Tree, Kruskal's Algorithms, Prim's Algorithms, Single Source Shortest Path, All pair Shortest Path, Maximum flow.

UNIT III (10 Hours)

Backtracking: General method, applications-n-queen problem, graph coloring, Hamiltonian cycles.

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution.

UNIT VI (8 Hours)

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP Complete classes, Cook's theorem.

Text and Reference Books:

1. Introduction to Algorithms, Thomas H CormenLeiserson et al, PHI, 2nd Edition 2001
2. Computer Algorithms: Introduction to Design and Analysis, Sara Baase and Allen Van Gelder, Pearson Education, 3rd Edition 2000
3. Algorithm Design, Jon Kleinberg and Eva Tardos, Pearson Education, 1st Edition 2005
4. The Design and analysis of Algorithms, A V Aho et al, Pearson Education, 3rd Edition 2007
5. Fundamentals of computer Algorithm, Ellis Horowitz, Sartaj Sahni and Rajasekharam, Galgotia Publication 2009

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand and explain algorithmic strategies such as divide and conquer, greedy, dynamic programming, backtracking, and branch and bound.
CO2: Analyze the time and space complexity of algorithms using asymptotic notations like Big O, Big Ω, and Big Θ.
CO3: Design efficient algorithms for a wide range of problems, optimizing for both performance and resource usage.
CO4: Apply algorithmic techniques to solve real-world computing problems and evaluate the effectiveness of different approaches.
CO5: Demonstrate knowledge of NP-completeness, and classify problems as tractable or intractable using reductions and complexity theory.

MCA413: Block Chain Technology

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Advanced Computer Networks, Cryptography and Network Security.

Course Objectives:

1. To introduce the fundamental concepts and architecture of blockchain technology, including blocks, cryptographic hash functions, and distributed ledgers.
2. To explain the working of consensus mechanisms such as Proof of Work, Proof of Stake, and their role in maintaining integrity in decentralized systems.
3. To familiarize students with blockchain platforms like Bitcoin, Ethereum, and smart contract development using tools such as Solidity.
4. To enable students to analyze and implement real-world blockchain applications in finance, supply chain, healthcare, and other industries.

Detailed Syllabus:

Unit-1

Introduction to Blockchain: History of centralized services, trusted third party for transactions, understand the difference between centralized, decentralized and distributed peer to peer networks, why Block chain? Types of Blockchain.

History of Bitcoins: How and when Blockchain and Bitcoin started. Milestone on the development of bitcoin, Problem area of Bitcoin, relation to Bitcoin, requirement of block chain in a business environment, sharing economy, requirements deep dive, Internet of value.

Unit-2

Consensus: Mechanism, Types of Consensus Mechanism, Consensus in Blockchain. Decentralization: Disintermediation and Contest Driven Decentralization, Routes to Decentralization, Full Ecosystem Decentralization, Smart Contracts, Decentralized Organizations, Platforms for Decentralization.

Unit-3

Blockchain Applications and USE case: Business drivers of blockchain, Digital currency and finance (including ICOs and alternative funding), Identity, Supply Chain, Healthcare, Ownership and property rights Governance and compliance.

Unit-4

Blockchain Challenges and Constraints: Blockchain risks, Technological challenges, standards Scalability issues, Security and privacy, Legal and regulatory problems, Social and cultural constraints.

Unit-5

Ethereum: Ethereum network, EVM, Transaction fee, Mist, Ether, gas, Solidity - Smart contracts, Truffle, Web3, Design and issue Cryptocurrency, Mining, DApps, DAO.

Unit-6

Introduction to Hyperledger Fabric: What is Hyperledger, Why Hyperledger, where can Hyperledger be used, Hyperledger Architecture, Membership, Blockchain, Transaction, Chain code, Hyperledger Fabric, Features of Hyperledger, prerequisite of Fabric installation

Suggested Readings:

1. A. Narayanan, J. Bonneau, E. Felten, A. Miller & S. Gold feder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
2. B. Singhal & G. Dhameja Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions, Apress 2018.
3. D. Mohanty, Blockchain - From Concept to Execution, BPB Publications, 2018.
4. Imran Bashir, Mastering Blockchain, 2nd Edition, Packt Publishing, 2018.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental principles, architecture, and components of blockchain systems.
CO2: Analyze different consensus mechanisms (e.g., Proof of Work, Proof of Stake) and their role in securing blockchain networks.
CO3: Develop and deploy smart contracts using platforms like Ethereum and programming languages such as Solidity.
CO4: Evaluate the applicability of blockchain in various sectors such as finance, supply chain, healthcare, and digital identity.
CO5: Demonstrate the ability to design simple decentralized applications (DApps) and understand key challenges in blockchain scalability and security.

MCA414: Cloud Computing and Virtualization

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Operating Systems, Computer Networking.

Course Objectives:

1. To introduce the fundamental concepts of cloud computing, including service models (IaaS, PaaS, SaaS), deployment models (public, private, hybrid), and cloud architecture.
2. To explain the principles of virtualization technologies, hypervisors, and their role in enabling scalable cloud infrastructures.
3. To familiarize students with cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, and their practical applications.
4. To enable students to design, develop, and deploy cloud-based and virtualized solutions, while addressing key issues like security, performance, and cost optimization.

Detailed Syllabus

Unit-1

Recent trends in computing: Introduction to Grid Computing: Motivation, Definition of Grid Computing, Evolution of Grid, Examples and Usages, Research Possibilities, Benefits of Grid Computing. Cluster Computing, Grid Computing, Utility Computing, Cloud Computing. Introduction to Grid Computing

Unit-2

Cloud Computing Fundamentals: Cloud Computing definition, Types of cloud, Cloud services: Benefits and challenges of cloud computing, Evolution of Cloud Computing, Applications cloud computing, Business models around Cloud – Major Players in Cloud Computing - Issues in Cloud - Eucalyptus - Nimbus - Open Nebula, Cloud Sim.

Unit-3

Cloud Computing Service Models: Infrastructure as a Service; Platform as a Service; Software as a Service. Accessing the Cloud: Web Applications, Web APIs, and Web Browsers.

Unit-4

Cloud Storage and Security: Overview, Advantages, Storage as a Service, Security, Reliability, Advantages, Cautions, Theft, Cloud Storage Providers. Standards: Applications, Client, Infrastructure, Services.

Unit-5

Virtualization Technologies: Basics of Virtualization - Types of Virtualizations - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Virtual Clusters and Resource management – Virtualization for Data-center Automation. Introduction to MapReduce, GFS, HDFS, Hadoop Framework.

UNIT-6

Security in the Cloud: Security Overview – Cloud Security Challenges and Risks – Software-as-a-Service Security – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control – Autonomic Security.

Text and Reference Books

1. The Grid- Blueprint for a New Computing Infrastructure, Ian Foster, Carl Kesselman, 2nd Edition, Morgan Kaufmann Publications,2003.
2. Grid Computing: Making the Global Infrastructure a Reality, Francine Berman, Geoffrey Fox, Tony Hey, John Wiley & Sons, 2003.
3. Cloud Computing: Principles and Paradigms, Rajkumar Buyya and James Broberg, John Wiley & Sons, 2011.
4. Cloud Computing, A Practical Approach, Anthony T Velte, Mc Graw Hill, 2010.

Course Outcomes:

Students will able to:

CO1: Understand the core concepts, architecture, and service models of cloud computing and its deployment strategies.

CO2: Analyze and apply virtualization technologies, including types of hypervisors, virtual machines, and containerization.

CO3: Demonstrate the ability to use cloud platforms (e.g., AWS, Azure, Google Cloud) for deploying and managing scalable cloud-based applications.

CO4: Identify and address key challenges in cloud computing, such as data security, privacy, service level agreements (SLAs), and performance.

CO5: Design and implement basic cloud and virtualized solutions to solve real-world computing problems effectively and efficiently.

MCA415: Big Data Analysis

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Database Management System, Data Mining and Warehousing.

Course Objectives:

1. To introduce the fundamental concepts, characteristics, and challenges of big data and its significance in modern data-driven decision-making.
2. To provide knowledge of big data frameworks and tools such as Hadoop, Spark, and MapReduce for processing and managing large-scale datasets.
3. To develop the ability to design and implement big data analytics solutions using data mining, machine learning, and statistical techniques.
4. To enable students to analyze real-world data sets, extract insights, and apply big data solutions in domains like business, healthcare, and social media.

Detailed Syllabus

UNIT I (6 Hours):
Introduction to Big Data Classification of Digital Data, Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics, Classification of Analytics, Top Challenges Facing Big Data, Responsibilities of data scientists, Big data applications in healthcare, medicine, advertising.

UNIT II (6 Hours):
Hadoop Architecture Hadoop Architecture, Hadoop Storage: HDFS, Hadoop MapReduce paradigm, Introduction to Hive, Introduction to Pig.

UNIT III (6 Hours):
Introduction to NoSQL & Hadoop Introduction to NoSQL Advantages of NoSQL, SQL versus No SQL, Introduction to Hadoop, Features of Hadoop, Hadoop Versions, Hadoop Versus SQL.

UNIT-IV (8 Hours):
Types of Analytics & Techniques Open-source technology for Big Data Analytics – cloud and Big Data – Mobile Business Intelligence and Big Data.

UNIT V (8 Hours):
Predictive Analysis Predictive Analytics, Supervised, Unsupervised learning, Clustering Techniques.

UNIT VI (6 Hours):
Basics of R, Working of R - Creating, listing and deleting the objects in memory - The on-line help Data with R Objects, R data Frames and Matrices, reading data in a file, saving data, generating data, manipulating data using R

Text and Reference Books

1. An Introduction to Statistical Learning: With Applications in R: Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.
2. BIG Data and Analytics, Sima Acharya, Subhashini Chhellappan, Willey
3. Vignesh Prajapati, "Big Data Analytics with R and Hadoop", Packet Publishing 2013.
4. The Culture of Big Data, Mike Barlow, by Oreilly
5. Big Data Analytics; Frank J. Ohlhorst, by Wiley

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the characteristics, architecture, and technologies associated with big data and its ecosystem.
CO2: Apply tools like Hadoop, Spark, and MapReduce to efficiently process and analyze large volumes of data.
CO3: Design and implement big data solutions using appropriate data models, NoSQL databases, and distributed computing techniques.
CO4: Evaluate and apply data analytics techniques, including data mining, machine learning, and real-time stream processing.
CO5: Solve complex real-world problems by extracting insights from big datasets and presenting them using visualization tools.

MCA416: Advanced Soft Computing						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	0	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Pre-requisites: Mathematics, Algorithm, Programming skills.

Course Objectives:

1. To introduce the fundamental principles of soft computing techniques such as fuzzy logic, neural networks, and evolutionary algorithms.
2. To provide an understanding of hybrid intelligent systems and their applications in solving complex and uncertain real-world problems.
3. To develop the ability to design, implement, and evaluate soft computing models for optimization, classification, and decision-making tasks.
4. To enable students to apply soft computing tools in various domains like robotics, data mining, image processing, and control systems.

Detailed Syllabus

Unit-1

Fuzzy Set Theory: Introduction to Neuro–Fuzzy and Soft Computing, Fuzzy Sets – Basic Definition and Terminology, Set-theoretic Operations, Member Function Formulation and Parameterization, Fuzzy Rules – Extension Principle and Fuzzy Relations, Fuzzy IF-THEN Rules, Fuzzy Reasoning – Fuzzy Inference Systems, Mamdani Fuzzy Models, Sugeno Fuzzy Models, Tsukamoto Fuzzy Models, Input Space Partitioning and Fuzzy Modeling.

Unit-2

Optimization: Derivative-based Optimization, Descent Methods – The Method of Steepest Descent, Classical Newton's Method, Step Size Determination, Derivative-free Optimization.

Unit-3

Genetic Algorithm: Simple Genetic Algorithms, Simulated Annealing, Gradient Free Optimization, Crossover and mutation, Genetic algorithms in search and optimization, Random Search, Downhill Simplex Search.

Unit-4

Neural Networks: Introduction, Architecture, Back Propagation and Feed Forward Networks, Supervised Learning Neural Networks, Perceptron, Adline, Back propagation Multilayer Perceptron, Radical Basis Function Networks, Unsupervised Learning Neural Networks, Competitive Learning Networks, Kohonen Self-Organizing Networks, Learning Vector Quantization, Hebbian Learning.

Unit-5

Neuro Fuzzy Modeling: Adaptive Neuro-Fuzzy Inference Systems, Architecture, Hybrid Learning Algorithm, Learning Methods that Cross-fertilize ANFIS and RBFN, Coactive Neuro Fuzzy Modeling, Framework Neuron Functions for Adaptive Networks, Neuro Fuzzy Spectrum.

Unit-6

Applications: Pattern Recognitions, Inverse Kinematics Problems, Automobile Fuel Efficiency Prediction, Image Processing, Biological Sequence Alignment and Drug Design, Robotics and Sensors, Information Retrieval Systems, Share Market Analysis, Natural Language Processing

Text and Reference Books

1. Neuro-Fuzzy and Soft Computing", J.S.R.Jang, C.T.Sun and E.Mizutani, PHI - Pearson Education, 2004.
2. Fuzzy Logic with Engineering Applications", Timothy J.Ross, McGraw-Hill, 1997.
3. Genetic Algorithms: Search, Optimization and Machine Learning", Davis E. Goldberg, Addison Wesley, N.Y., 1989.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand and explain the core concepts of fuzzy logic, artificial neural networks, and genetic algorithms.
CO2: Apply fuzzy inference systems and neural network architectures to solve classification and prediction problems.
CO3: Design and implement hybrid soft computing systems by integrating multiple techniques to enhance problem-solving efficiency.
CO4: Analyze and compare various soft computing models based on accuracy, convergence, and applicability to real-world problems.
CO5: Use modern tools and programming environments (e.g., MATLAB, Python libraries) to simulate and evaluate soft computing solutions in domains like robotics, bioinformatics, and intelligent systems.

MCA417: Data Science

L	T	P	Theory	Internal	Practical	Total Marks
Hours						
3	1	0	70	30	4	100

Teaching Scheme	Examination Scheme
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Pre-requisites: Basic Knowledge of Programming, Understanding of Mathematics and Statistics:

Course objectives:

1. To provide a comprehensive understanding of data science concepts, tools, and methodologies for collecting, processing, analyzing, and visualizing data.
2. To develop the ability to apply statistical methods and machine learning algorithms to extract meaningful patterns and predictions from large datasets.
3. To enable students to work with real-world data using data wrangling, data cleaning, and exploratory data analysis techniques in Python or R.
4. To equip learners with practical skills to design and deploy data-driven solutions in various domains like business intelligence, healthcare, finance, and social media analytics.

Unit 1:

Data Science - An Overview: Introduction to Data Science, Definition and description, history and development, terminologies, basic framework and architecture, difference between Data Science and business analytics, importance of Data Science, primary components of Data Science, users of Data Science and its hierarchy, overview of Data Science techniques, challenges and opportunities in Data Science, industrial application of Data Science techniques.

Unit 2:

Mathematics and Statistics in Data Science: Role of mathematics, importance of probability and statistics, important types of statistical measures: Descriptive, Predictive and prescriptive statistics, introduction to statistical inference, application of statistical techniques, linear algebra: matrix and vector theory, role of linear algebra in Data Science, exploratory data analysis and visualization techniques, difference between exploratory and descriptive statistics.

Unit 3:

Machine Learning in Data Science: Role of machine learning, different types of machine learning techniques and its broad scope: Supervised, unsupervised, reinforcement and deep learning, difference between different machine learning techniques, machine learning algorithms, importance of machine learning in today's business, difference between classification and prediction.

Unit 4:

Computers in Data Science: Role of computer science in Data Science, various components of computer science being used for Data Science, role of relation data base systems: SQL, NoSQL, data warehousing, importance of operating concepts and memory management, freely available software tools: R, Python, important proprietary software tools, business intelligence tools.

Unit 5:

Data Science Project Management: Data Science project framework, execution flow of a Data Science project, various components of Data Science projects, stakeholders of Data Science project, industry use cases of Data Science implementation, challenges and scope of Data Science project management, process evaluation model, comparison of Data Science project methods, improvement in success of Data Science project models.

Unit 6:

Case Studies of Data Science Application: Weather forecasting, Stock market prediction, Object recognition, Real Time Sentiment Analysis

Text Books:

1. Doing Data Science, Straight Talk from The Frontline. Cathy O'Neil and Rachel Schutt, O'Reilly, 2014.
2. Jiawei Han, Micheline Kamber and Jian Pei. Data Mining: Concepts and Techniques, 3rd ed. The Morgan Kaufmann Series in Data Management Systems.
3. K G Srinivas, G M Siddesh, "Statistical programming in R", Oxford Publications.

Reference Book:

1. Introduction to Data Mining, Pang-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.
2. Brain S. Everitt, "A Handbook of Statistical Analysis Using R", Second Edition, 4 LLC, 2014.
3. Dalgaard, Peter, "Introductory statistics with R", Springer Science & Business Media, 2008.
4. Paul Teetor, "R Cookbook", O'Reilly, 2011.

Course Outcome:

After completing this course, students will be able to:

CO1: Understand and explain the core concepts of data science, including data preprocessing, data visualization, and statistical analysis.
CO2: Apply data science tools and programming languages such as Python or R to collect, clean, analyze, and visualize data.
CO3: Develop and evaluate machine learning models for classification, regression, and clustering tasks using real-world datasets.
CO4: Interpret analytical results and generate actionable insights to support decision-making in various domains.
CO5: Demonstrate the ability to manage data science projects and communicate findings effectively through visual and written reports.