Semester-III

S No	Course Code	Course	Course Type	L	T	P	Credits
1.	CAPC1322M	Data Structures using C	Major	4	0	2	6
2.	ACPC1322N	Web Development using HTML, CSS and JS	Minor	4	0	2	6
3.	CAP022I	Introduction to Computers *(offered for other Majors)*	MD	2	1	0	3
4.	WDP322S	Web Developer-III: Introduction to PHP and MYSQL*(offered for other Majors and Computer Applications)*	SEC	2	0	2	2
5.	DTS022	VAC (*for BCA from other Majors*) Digital and Technological Solutions*(offered for other Majors)*	VAC	2	0	0	2
6.	CNS022A/ URM022A/ ENL022A	Communication skill / Modern Indian language / English Language*(offered by other Departments)*	AEC	3	0	3	3

SEMESTER-IV

S No	Course Code	Course	Course Type	L	T	P	Credits
1.	CAPC1422M	Object Oriented Programming withJAVA	Major	4	0	2	6
2.	CAPC2422M	Computer Organization & Architecture	Major	4	0	2	6
3.	CAPC3422M	Data Communication and Networks	Major	4	0	2	6
4.	ACPC1422N	Introduction to Database Systems	Minor	4	0	2	6

Course Type: - Major
Paper Title: - **DATA STRUCTURES USING C**
Credit Weightage: - THEORY -04; PRACTICALS- 02

Semester: - 3rd
Paper Code:- CAPC1322M
Batch: - 2023

Course Objective:

- To understand the need and significance of Data structures as a computer Professional.
- To teach concept and implementation of linear and nonlinear data structures.
- Introduces a variety of data structures such as stack, queue,
- hash tables, search trees, heaps, graphs.
- To introduce various techniques for representation of the data in the real world.
- Introduces sorting and pattern matching algorithms.

Course Outcomes:

- Ability to select the data structures that efficiently model the information in a problem.
- Ability to assess efficiency trade-offs among different data structure implementations or combinations.
- Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, heaps, graphs.
- Implement and know the application of algorithms for sorting and pattern matching.

UNIT – I

Introduction to Data Structure and Algorithms- Introduction: Structured programming concepts, Basic concept of data structures and pointers. Dynamic memory management in C. Fundamentals of algorithm analysis, Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth, Algorithm efficiency – best case, worst case and average case.

Linear Data Structures using Arrays. Abstract Data-types: 1D and 2D array, Stack - Applications of stack: Expression Evaluation - Conversion of Infix to postfix and prefix expression, Tower of Hanoi.

Queue - Types of Queues: Circular Queue, Double Ended Queue, Applications – Priority Queue using Arrays

UNIT – II

Linked Lists: Linked List and its comparison with array implementation. Types of Linked lists, Applications of Linked lists. Implementing Linked Lists using structures. Insertion, Deletion, Search, display and reverse in single and doubly linked lists.

Linked List implementation of Stack, Queues, Circular Queues. Polynomial manipulation using Linked lists.

UNIT – III

Sorting and Searching- Searching - Linear Search and binary search, Applications - Finding square root of 'n'-Longest Common Prefix Sorting – Insertion sort - Selection sort – Bubble sort – (Counting Sort) - Quick sort- Merge sort, Shell Sort, radix sort, Heap sort algorithms and their Analysis.

File Organization - Organization (Sequential, Direct, Indexed Sequential, Hashed) and various types of accessing schemes.

UNIT – IV

Non- linear Data Structures - Graph Algorithms: Graphs and their Representations- Adjacency list and adjacency matrix representations. Basic graph definitions: paths, cycles, trees, spanning trees, connected components, Euler's formula. Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths.

Trees: Terminology, Binary Tree – Terminology and Properties, Tree Traversals, Expression Trees – Binary Search Trees – operations in BST – insertion, deletion, finding min and max. AVL trees and rotations in AVL trees.

Hashing - Hash functions, open hashing and closed hashing techniques.

TEXT& REFERENCES:

1. Data Structures using C – A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.
2. Data Structures and Algorithms Made Easy, Narasimha Karumanchi, Career Monk.
3. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.
4. Data Structures: A Pseudocode Approach with C, 2 nd Edition, R. F. Gilberg and B.A.Forouzan,Cengage Learning.
5. Websites like VisuAlgo(https://visualgo.net/) and Data Structure Visualizations (www.cs.usfca.edu/~galles/visualization) provide interactive visualizations of various data structures and algorithms.

LAB WORK - DATA STRUCTURES (CAPC1322M)

LIST OF DATA STRUCTURE IMPLEMENTATIONS:

1. Write a program that implements stack and its operations using Arrays and Pointers.
2. Write a program that implements queue and its operations using Arrays and Pointers.
3. Write a program that uses functions to perform Creation, Insertion, Deletion and Traversal operations on a singly linked list.
4. Write a Program to implement various Hash table operations.
5. Write a program that uses functions to perform Creation, Insertion, Deletion and Traversal operations on a doubly linked list.
6. Write a program that uses functions to perform Creation, Insertion, Deletion and Traversal operations on a circular linked list.
7. Write a program to implement iterative and recursive tree traversal methods (preorder, inorder, postorder).
8. Write a program to implement Binary Search tree, Tree.
9. Write a program to implement BFS/DFS graph traversal methods.
10. Write a Program to implement Hashing techniques.

Course Type: - Minor
Paper Title: - **WEB DEVELOPMENT USING HTML, CSS AND JS**
Credit Weightage: - THEORY -04; PRACTICALS- 02

Semester: - 3rd
Paper Code: - ACPC1322N
Batch: - 2023

Course Objective:

- To introduce the fundamentals of web development.
- To provide a comprehensive understanding of HTML, CSS, and JavaScript.
- To enable students to style web pages using CSS for enhanced visual appeal.
- To enable students to build responsive and mobile-friendly web pages.

Course Outcomes:

- Understand the structure and components of a web page.
- Apply CSS styles to control the layout, typography, and visual presentation of web pages.
- Write JavaScript code to manipulate the Document Object Model (DOM) and handle user events.
- Create a complete and functional website using HTML, CSS, and JavaScript.

UNIT – I

Introduction to HTML: Mark-up Languages, Introduction to HTML5, Development Environment Setup, Anatomy of an HTML Tag, BasicStructure of HTML Document, HTML Content Models, Meta-Tags, Formatting Tags, Text Level Formatting, Lists,Hyperlinks, Image and Image Maps, Table Tags, HTML Comment tag. Block and inline elements, redirecting to another URL, creating division-based layouts. Forms: creating basic form, using check boxes, textboxes and option buttons, input validation and additional input types in HTML5, HTML multimedia basics. HTML DOM structure.

UNIT – II

Cascading Style Sheets: Need for CSS. Different approaches to style sheets, Anatomy of a CSS Rule. CSS syntax, selectors, and properties. Element, Class, and ID Selectors. Combining Selectors, Pseudo-Class Selectors and conflict Resolution. Controlling typography with CSS: fonts, colours, and text formatting. Box model: margins, borders, padding, and dimensions. Positioning elements using CSS: static, relative, absolute, and fixed. Creating layouts with CSS: float, flexbox, and grid. Working with backgrounds, gradients, and shadows. Adding transitions and animations with CSS. Responsive design using CSS media queries. . Web fonts. Basic Introduction to Bootstrap Framework.

UNIT – III

Introduction to JavaScript: JavaScript basics its role in web development, Different approaches to place JavaScript code in an HTML File. JavaScript syntax, Variable Declaration and Assignment. Const, let and var. Expressions and Operators: Arithmetic, Relational, Logical, Assignment and Evaluation Expressions. Type Conversions. Control structures: conditional statements and loops. Functions: Defining, Invoking, Function Arguments and Parameters. Functions as Values. Event handling, Form Validation and interactivity. Working with JSON data and APIs. Basic Introduction to popular JavaScript libraries.

UNIT – IV

Objects and Classes in JavaScript: Creating Objects, Querying and setting Properties, Deleting and Testing Properties. Serializing Objects. Arrays: Creating, Reading, Writing arrays. Array length. Iterating Arrays, Strings as Arrays. Client-side storage: cookies and local storage. Introduction to version control systems for collaborative development. Emerging trends and future directions in web development.

TEXT & REFERENCES:

1. HTML and CSS: Design and Build Websites" by Jon Duckett
2. JavaScript and jQuery: Interactive Front-End Web Development" by Jon Duckett
3. Eloquent JavaScript: A Modern Introduction to Programming" by MarijnHaverbeke
4. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins
5. CSS: The Missing Manual" by David Sawyer McFarland
6. JavaScript: The Good Parts" by Douglas Crockford
7. JavaScript: The Definitive Guide" by David Flanagan
8. Mozilla Developer Network (MDN) web documentation: https://developer.mozilla.org/
9. W3Schools online tutorials: https://www.w3schools.com/
10. FreeCodeCamp's web development curriculum: https://www.freecodecamp.org/

LAB WORK - WEB DEVELOPMENT USING HTML, CSS AND JS (ACPC1322N)

LIST OF IMPLEMENTATIONS/PROGRAMES:

1. Create a simple HTML page with headings, paragraphs, and lists.
2. Embed images and videos into the HTML page.
3. Design a web page with links to different pages and allow navigation between web pages.
4. Use a HTML table to design a page with a header, sidebar, main content and footer.
5. Design a user registration form using different HTML form controls.
6. Style the HTML using CSS.
7. Customize fonts, colors, and backgrounds.
8. Apply CSS selectors to style specific elements.
9. Implement CSS animations for hover effects on buttons and images; add smooth transitions when hovering over navigation links.
10. Create a web page layout using CSS Flexbox to arrange elements in a flexible and responsive manner.
11. Build a card-based layout to display multiple content blocks with equal spacing.
12. Experiment with responsive typography by using relative units like em, rem, and vw to adapt font sizes to different screen sizes
13. Implement a responsive navigation bar using media queries.
14. Design a web page with buttons that can handle different page events using JS event handlers.
15. Use Java Script to change the image displayed in an img tag when a button on the page is clicked.
16. Develop a registration form with JavaScript form validation for input fields with appropriate error messages for invalid inputs.
17. Use bootstrap to add formatting to your home page.
18. Write a JavaScript program with proper GUI to perform unit conversion using the onChange event.
19. Design the interface of a login page using HTML and CSS.
20. Design a simple "To Do" Application using HTML/CSS/JavaScript.
21. Design Basic Calculator using HTML/CSS/JavaScript.
22. Design and develop a simple "Tic-Tac-Toe Game" using HTML/CSS/JavaScript.
23. Use JavaScript's Local Storage API to store and retrieve data from the browser.
24. Create a simple note-taking application that saves notes locally.
25. Design and develop a complete website using HTML, CSS, and JavaScript.

Course Type: - Major

Semester: - 4th

Paper Title: - **OBJECT ORIENTED PROGRAMMING WITH JAVA**

Paper Code: - CAPC1422M

Credit Weightage: - THEORY -04; PRACTICALS- 02

Batch: - 2023

Course Objective:

- To understand the basic object-oriented programming concepts and apply them in problem solving.
- To illustrate inheritance concepts for reusing the program.
- To demonstrate multitasking by using multiple threads and event handling.
- To understand the basics of java console and GUI based programming.

Course Outcomes:

- Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection.
- Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords.
- Use multithreading concepts to develop inter process communication.
- Understand the process of graphical user interface design and implementation using AWT /Swings.

UNIT – I

Java Fundamentals - Java Basics: Java Design goal - Features of Java Language - JVM - Bytecode - Java source file structure basic programming constructs- datatypes, variables, scope and life time of variables, operators, expressions, control statements, type conversion and casting, classes and objects, constructors, methods, access control, this keyword, garbage collection, Arrays- one dimensional and multi-dimensional arrays, arrays of objects and string handling.

UNIT – II

Object Oriented Programming- Class Fundamentals, Object reference, object initialization and constructors, parameterized and overloaded constructors, method overloading and method overriding, this reference, static block, nested classes, inner class, finalize () method, Wrapper classes, Inheritance types, use of super, polymorphism, abstract classes.

Packages and interfaces: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interfaces, applying interfaces, variables in interface and extending interfaces.

UNIT – III

Exception Handling and Concurrency - Exception Handling, Exceptions & Errors, Types of Exception, Control Flow in Exceptions, Use of try, catch, finally, throw and throws in Exception Handling. User defined exceptions.Exploring Java.util class – Collections, Lists, Maps and Sets.

Multithreading - Thread creation, sharing the workload among threads, synchronization inter thread communication deadlock.

UNIT – IV

GUI Programming, Event Handling and Database Connectivity

Applets: Concepts of Applets, life cycle of an applet, types of applets, creating applets, passing parameters to applets. **Event Handling**: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events.

AWT & SWINGS: The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager –border, grid, flow, card and grid bag layout. limitations of AWT, MVC architecture. Introduction to JavaFx and Java Swings.

JDBC: JDBC Drivers, JDBC API.

TEXT & REFERENCES:

1. Java the complete reference, 7th edition, Herbert schildt, Tata McGraw Hill.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.
3. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, 8th Edition, Pearson Education.
4. Programming with Java, E. Balagurusamy, Tata McGraw Hill.
5. Java: How to Program P.J.Deitel and H.M.Deitel, PHI.

LAB. -Object Oriented Programming through Java (CAPC1422M)

LIST OF Programs:

1. Write a Java program to print "Hello, World!" on the console.
2. Create a program that takes two numbers as input and displays their sum, difference, product, and quotient.
3. Implement a program to check whether a given number is prime or not.
4. Develop a Java program to calculate the factorial of a given number using both iterative and recursive approaches.
5. Create a class representing a "Student" with attributes like name, roll number, and age. Include methods to display student details.
6. Design a "Rectangle" class with attributes length and width. Implement methods to calculate area and perimeter.
7. Develop a "BankAccount" class with attributes account number and balance. Include methods to deposit, withdraw, and display the balance.
8. Create a superclass "Shape" with attributes color and area. Derive subclasses like "Circle," "Square," and "Triangle" from the "Shape" class. Implement methods to calculate the area for each shape.
9. Design a "Vehicle" class with methods like start(), stop(), and accelerate(). Extend this class to create "Car" and "Motorcycle" subclasses with additional methods specific to each vehicle type.
10. Create an abstract class "Animal" with abstract methods like "sound()" and "eat()". Implement subclasses like "Dog," "Cat," and "Bird" with specific sound and eat behaviors.
11. Design an interface "Playable" with a method "playSound()". Implement classes like "Guitar," "Piano," and "Drums" that implement the "Playable" interface.
12. Implement a program that reads data from a file and performs some calculations. Handle file-related exceptions like file not found or I/O errors.
13. Implement a program to store a list of names in an ArrayList and perform operations like add, remove, and search for a name.
14. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.
15. Create a HashMap to store student names and their corresponding marks. Display the student with the highest mark
16. Create a simple calculator GUI with buttons for numbers and operations (+, -, *, /).

Course Type: - Major
Paper Title: - **COMPUTER ORGANIZATION & ARCHITECTURE**
Credit Weightage: - THEORY -04; PRACTICALS- 02

Semester: - 4th
Paper Code: - CAPC2422M
Batch: - 2023

Course Objective:

- To introduce principles of computer organization and the basic architectural concepts.
- Topics include computer arithmetic, instruction set design, micro programmed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.

Course Outcomes:

- Understand the basics of instruction sets and their impact on processor design.
- Demonstrate an understanding of the design of the functional units of a digital computer system.
- Recognize and manipulate representations of numbers stored in digital computers.

UNIT – I

DIGITAL COMPUTERS: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture. Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit. Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT – II

MICRO PROGRAMMED CONTROL: Control memory, Address sequencing, micro program example, design of control unit. Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT – III

DATA REPRESENTATION: Data types, Complements, Fixed Point Representation, Floating Point Representation. Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating point Arithmetic operations.

UNIT – IV

INPUT-OUTPUT ORGANIZATION: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access. Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics. Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

TEXT & REFERENCES:

1. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.
2. Computer Organization – Carl Hamacher, ZvonksVranesic, SafeaZaky, McGraw-Hill.
3. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
4. Structured Computer Organization – Andrew S. Tanenbaum, PHI/Pearson.

LAB WORK - Computer Organization & Architecture (CAPC2422M)

LIST OF EXPERIMENTS:

1. Simulations of Control Unit.....

Course Type: - Major
Paper Title: - DATA COMMUNICATION AND NETWORKS
Credit Weightage: - THEORY -04; PRACTICALS- 02

Semester: - 4th
Paper Code: - CAPC3422M
Batch: - 2023

Course Objective:

- Equip the students with a general overview of the concepts and fundamentals of computer networks.
- Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

Course Outcomes:

- Gain the knowledge of the basic computer network technology.
- Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
- Obtain the skills of subnetting and routing mechanisms.
- Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT – I

Introduction: Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission and transmission Impairments. Data link layer: Design issues, framing, Error detection and correction.

UNIT – II

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channels.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT – III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control, Quality of Service, Internetworking, the Network layer in the internet.

UNIT – IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

Application Layer: Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

TEXT & REFERENCES:

1. Data Communications and Networking – Behrouz A. Forouzan. Third, Tata McGraw Hill.
2. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI.
3. Computer Networking: A Top-Down Approach, 6th Edition, Pearson.

LAB. DATA COMMUNICATION AND NETWORKS (CAPC3422M)

LIST OF EXPERIMENTS:

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting technique used in buffers.
10. Perform the following using NS2 Simulator
    - I. NS2 Simulator-Introduction
    - II. Simulate to Find the Number of Packets Dropped
    - III. Simulate to Find the Number of Packets Dropped by TCP/UDP
    - IV. Simulate to Find the Number of Packets Dropped due to Congestion
    - V. Simulate to Compare Data Rate & Throughput.
    - VI. Simulate to Plot Congestion for Different Source/Destination
    - VII. Simulate to Determine the Performance with respect to Transmission of Packets .....

Course Type: - Minor
Paper Title: - **INTRODUCTION TO DATABASE SYSTEMS**
Credit Weightage: - THEORY -04; PRACTICALS- 02

Semester: - 4th
Paper Code: - ACPC1422N
Batch: - 2023

Course Objective:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- To familiarize with data models, database design, relational model, relational algebra, transaction control, concurrency control.

Course Outcomes:

- Gain knowledge of fundamentals of DBMS, database design and normal forms.
- Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.

UNIT – I

DATABASE SYSTEM APPLICATIONS: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS.
INTRODUCTION TO DATABASE DESIGN: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model.

UNIT – II

INTRODUCTION TO THE RELATIONAL MODEL: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views. Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT – III

SQL: Queries, Constraints, Triggers: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases. SCHEMA REFINEMENT: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies.

UNIT – IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

TEXT & REFERENCES:

1. Database System Concepts, Silberschatz, Korth, Mc Graw Hill, 7th Edition.
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill.
3. Fundamentals of Database Systems, ElmasriNavrate, Pearson Education.
4. Introduction to Database Systems, C. J. Date, Pearson Education.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. SQL, PL/SQL – The Programming Language of Oracle, Ivan Bayross, BPB Publications.

LAB WORK - INTRODUCTION TO DATABASE SYSTEMS (ACPC1422N)

LIST OF IMPLEMENTATIONS/PROGRAMS:

Implement the following on a given application:

1. Concept design with E-R Model.
2. Relational Model.
3. Normalization.
4. Practicing DDL commands.
5. Practicing DML commands.
6. Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger).
9. Stored Procedures.
10. Usage of Cursors.