

ANDHRA KESARI UNIVERSITY



B.Sc., Honours in Computer Science MAJOR

w.e.f AY 2023-24 onwards

COURSE STRUCTURE

Year	Semester	Course	Title	Hrs./ Week	Credits	
I	I	1	Essentials and applications of Mathematical, Physical and Chemical Sciences	5	4	
		2	Advances in Mathematical, Physical and Chemical Sciences	5	4	
	II	3	3	Problem Solving using C - (T)	3	3
			3	Problem Solving using C- (P)	2	1
		4	4	Digital Logic Design- (T)	3	3
			4	Digital Logic Design- (P)	2	1
II	III	5	5	Object Oriented Programming using Java- (T)	3	3
			5	Object Oriented Programming using Java - (P)	2	1
		6	6	Data Structures using C - (T)	3	3
			6	Data Structures using C - (P)	2	1
		7	7	Computer Organization - (T)	3	3
			7	Computer Organization- (P)	2	1
	8	8	Operating Systems - (T)	3	3	
		8	Operating Systems - (P)	2	1	
	IV	9	9	Database Management System - (T)	3	3
			9	Database Management System - (P)	2	1
		10	10	Object Oriented Software Engineering - (T)	3	3
			10	Object Oriented Software Engineering - (P)	2	1
		11	11	Data Communications and Computer Networks - (T)	3	3
			11	Data Communications and Computer Networks - (P)	2	1

Semester- III
Course 5: Object Oriented Programming using Java

Credits -3

Course Objectives

To introduce the fundamental concepts of Object-Oriented programming and to design & implement object-oriented programming concepts in Java.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand the basic concepts of Object-Oriented Programming and Java Program Constructs
2. Implement classes and objects and analyze Inheritance and Dynamic Method Dispatch
3. Demonstrate various classes in different packages and can design own packages
4. Manage Exceptions and Apply Threads
5. Create GUI screens along with event handling

UNIT-I

OOPs Concepts and Java Programming: Introduction to Object-Oriented concepts, procedural and object-oriented programming paradigm

Java programming: An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting, Accepting Input from the Keyboard, Reading Input with `Java.util.Scanner` Class, Displaying Output with `System.out.printf()`, Displaying Formatted Output with `String.format()`, Control Statements

UNIT-II

Arrays, Command Line Arguments, Strings-String Class Methods

Classes & Objects: Creating Classes, declaring objects, Methods, parameter passing, static fields and methods, Constructors, and 'this' keyword, overloading methods and access

Inheritance: Inheritance hierarchies, super and subclasses, member access rules, 'super' keyword, preventing inheritance: final classes and methods, the object class and its methods; **Polymorphism:** Dynamic binding, method overriding, abstract classes and methods;

UNIT-III

Interface: Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;

Packages: Defining, creating and accessing a package, understanding `CLASSPATH`, importing packages.

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

UNIT-IV

Multithreading: Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization

UNIT-V

GUI Programming with Swing- Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, GridBag Layout.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books

1. Cay S. Horstmann, “Core Java Fundamentals”, Volume 1, 11 th Edition, Prentice Hall, 2018.
2. Paul Deitel, Harvey Deitel, “Java SE 8 for programmers”, 3rd Edition, Pearson, 2015.
3. S. Malhotra, S. Chudhary, Programming in Java, 2nd edition, Oxford Univ. Press.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on Object-Oriented Programming Concepts and Java Constructs

Evaluation Method: Quiz Performance and Knowledge Retention

Unit 2: Activity: Object-Oriented Programming Assignment: Class Implementation

Evaluation Method: Assignment Completion and Correctness

Unit 3: Activity: Hands-on Lab Activity: Creating and Using Custom Java Packages

Evaluation Method: Lab Performance and Correctness of Code Implementation

Unit 4: Activity: Case Study Discussion on where multi-threading is crucial

Evaluation Method: Critical thinking, problem-solving, and presentation skills.

Unit 5: Activity: GUI design contest using Java Swings

Evaluation Method: GUI design, Visual appearance and user friendliness, usability, and adherence to event handling principles.

III Semester
Course 5: Object Oriented Programming using Java Lab
Credits -1

List of Experiments

1. Write a Java program to print Fibonacci series using for loop.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
4. Write a Java program that implements method overloading.
5. Write a Java program for sorting a given list of names in ascending order.
6. Write a Java program that displays the number of characters, lines and words in a text file.
7. Write a Java program to implement various types of inheritance
 - i. Single
 - ii. Multi-Level
 - iii. Hierarchical
 - iv. Hybrid
8. Write a java program to implement runtime polymorphism.
9. Write a Java program which accepts withdraw amount from the user and throws an exception “In Sufficient Funds” when withdraw amount more than available amount.
10. Write a Java program to create three threads and that displays “good morning”, for every one second, “hello” for every 2 seconds and “welcome” for every 3 seconds by using extending Thread class.
11. Write a Java program that creates three threads. First thread displays “OOPS”, the second thread displays “Through” and the third thread Displays “JAVA” by using Runnable interface.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the key board is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button “add” is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form which having the following fields and button SAVE

Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

III Semester
Course 6: Data Structures using C
Credits -3

Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand various Data Structures for data storage and processing.
2. Realize Linked List Data Structure for various operations
3. Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures.
4. Understand and implement various searching & sorting techniques.
5. Understand the Non-Linear Data Structures such as Binary Trees and Graphs

UNIT-I

Basic Concepts: Pointers and dynamic memory allocation, Algorithm-Definition and characteristics, Algorithm Analysis-Space Complexity, Time Complexity, Asymptotic Notation **Introduction to Data structures:** Definition, Types of Data structure, Abstract Data Types (ADT), Difference between Abstract Data Types, Data Types, and Data Structures.

Arrays-Concept of Arrays, Single dimensional array, Two dimensional array, Operations on arrays with Algorithms (searching, traversing, inserting, deleting)

UNIT-II

Linked List: Concept of Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array, Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list;

Implementation of Linked List ADT: Creating a List, Traversing a linked list, Searching linkedlist, Insertion and deletion into linked list (At first Node, Specified Position, Last node), Application of linked lists

UNIT-III

Stacks: Introduction to stack ADT, Representation of stacks with array and Linked List, Implementation of stacks, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation - Tower of Hanoi, Recursion: Concept and Comparison between recursion and Iteration

Queues: Introduction to Queue ADT, Representation of Queues with array and Linked List, Implementation of Queues, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue

UNIT-IV

Searching: Linear or Sequential Search, Binary Search and Indexed Sequential Search

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

UNIT-V

Binary Trees: Concept of Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree.

Graphs: Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs (DFS, BFS), Application of Graphs.

Text Books:

1. Horowitz and Sahani, “Fundamentals of Data Structures”, Galgotia Publications Pvt Ltd Delhi India.
2. A.K. Sharma ,Data Structure Using C, Pearson Education India.
3. “Data Structures Using C” Balagurusamy E. TMH

Reference Books

1. “Data Structures through C”, Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, “Data Structure Using C and C++” Wiley Dreamtech Publication.
3. Lipschutz, “Data Structures” Schaum’s Outline Series, Tata Mcgraw-hill Education (India)Pvt. Ltd .
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount “Data Structures and Algorithms in C++”, Wiley India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Algorithm analysis exercises

Evaluation Method: Programming Assignment and Correctness

Unit 2: Activity: Presentations on real-life applications of linked lists

Evaluation Method: Presentation skills or reports

Unit 3: Activity: Role-playing activities for stack operations

Evaluation Method: Problem-solving skills, communication and collaboration abilities.

Unit 4: Activity: Sorting algorithm analysis and comparison activities

Evaluation Method: Performance analysis and presentation.

Unit 5: Activity: Case Study on Applications of Graphs

Evaluation Method: Critical thinking, problem-solving, and presentation skills

III Semester
Course 6: Data Structures Using C
Credits -1

List of Experiments:

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array
 - a. Add an element at the beginning of an array
 - b. Insert an element at given index of array
 - c. Update an element using a values and index
 - d. Delete an existing element
2. Write Program to implement Single Linked List with insertion, deletion and traversal operations
3. Write Program to implement Circular doubly Linked List with insertion, deletion and traversal operations
4. Write Programs to implement the Stack operations using an array
5. Write a program using stacks to convert a given infix expression to postfix
6. Write Programs to implement the Stack operations using Liked List.
7. Write Programs to implement the Queue operations using an array.
8. Write Programs to implement the Queue operations using Liked List.
9. Write a program for Binary Search Tree Traversals
10. Write a program to search an item in a given list using the following Searching Algorithms
 - a. Linear Search
 - b. Binary Search.
11. Write a program for implementation of the following Sorting Algorithms
 - a. Bubble Sort
 - b. Insertion Sort
 - c. Quick Sort

III Semester
Course 7: Computer Organization
Credits -3

Course Objectives

To familiarize with organizational aspects of memory, processor and I/O.

Course Outcomes

Upon successful completion of the course, the students will be able to

1. Identify different types of instructions
2. Differentiate between micro-programmed and hard-wired control units.
3. Analyse the performance of hierarchical organization of memory.
4. Summarize different data transfer techniques.
5. Demonstrate arithmetic operations on fixed- and floating-point numbers and illustrate concepts of parallel processing.

UNIT – I

Register Transfer Language and Micro Operations: Introduction- Functional units, computer registers, register transfer language, register transfer, bus and memory transfers, arithmetic, logic and shift micro-operations, arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, instruction cycle.

Register reference instructions, Memory – reference instructions, input – output and interrupt.

UNIT – II

CPU and Micro Programmed Control: Central Processing unit: Introduction, instruction formats, addressing modes. Control memory, address sequencing, design of control unit - hard wired control, micro programmed control.

UNIT – III

Memory Organization: Memory hierarchy, main memory, auxiliary memory, associative memory, cache Memory and mappings.

UNIT – IV

Input-Output Organization: Peripheral Devices, input-output interface, asynchronous data transfer, modes of transfer- programmed I/O, priority interrupt, direct memory access, Input – Output Processor (IOP).

UNIT – V

Computer Arithmetic and Parallel Processing: Data representation- fixed point, floating point, addition and subtraction, multiplication and division algorithms.

Parallel Processing-Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline.

Text Books:

1. M. Moris Mano, “Computer Systems Architecture”, 3rd edition, Pearson/ PHI.

Reference Books:

1. Carl Hamacher, ZvonksVranesic, SafeaZaky, “Computer Organization”, 5th edition, McGraw Hill.
2. William Stallings, “Computer Organization and Architecture”, 8th edition, Pearson/PHI.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz competition on micro-operations.

Evaluation Method: Accuracy and speed in answering quiz questions.

Unit 2: Activity: Instruction Format Puzzle: Solving a puzzle to decode and understand instruction formats.

Evaluation Method: Accuracy and speed in completing the puzzle.

Unit 3: Activity: Memory Hierarchy Poster: Creating informative posters or infographics on memory hierarchy.

Evaluation Method: Clarity of information, presentation and creativity of visual design.

Unit 4: Activity: I/O Troubleshooting Challenge

Evaluation Method: problem identification, feasibility of proposed solutions, and clarity of explanations.

Unit 5: Activity: Case Study on Parallel processing architecture.

Evaluation Method: Understanding of parallel processing concepts and architectures.

III Semester

Course 3: Computer Organization

Credits -1

Lab Experiments

1. Implement a C program to convert a Hexadecimal, octal, and binary number to decimal number vice versa.
2. Implement a C program to perform Binary Addition & Subtraction.
3. Implement a C program to perform Multiplication of two binary numbers.
4. Implement arithmetic micro-operations using logic gates.
5. Implement logic and shift micro-operations using logic gates.
6. Implement a C program to perform Multiplication of two binary numbers (signed) using Booth's Algorithms.
7. Implement a C program to perform division of two binary numbers (Unsigned) using restoring division algorithm.
8. Implement a C program to perform division of two binary numbers (Unsigned) using non-restoring division algorithm.
9. Write assembly language code for $A+B*(C-D)$ using various instruction formats in MASM or any open-source assembler.
10. Write assembly language code for $A+B*C$ using various addressing modes in MASM or any open-source assembler.

III Semester
Course 8: Operating Systems
Credits -3

Course Objectives

To gain knowledge about various functions of an operating system like memory management, process management, device management, etc.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Demonstrate knowledge and comprehension of operating system functions.
2. Analyze different process scheduling algorithms and apply them to manage processes and threads effectively
3. Create strategies to prevent, detect, and recover from deadlocks, and design solutions for inter-process communication and synchronization problems.
4. Compare and contrast different memory allocation strategies and evaluate their effectiveness
5. Evaluate disk scheduling algorithms while implementing OS security measures

UNIT- I

What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT- II

Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling- Non-Preemptive and Preemptive Scheduling Algorithms.

UNIT III

Process Management: Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT IV

Memory Management: Physical and Virtual Address Space; Memory Allocation Strategies–Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT V

File and I/O Management, OS security: Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling algorithms.

Text Books:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne (7th Edition) Wiley India Edition.

Reference Books

1. Operating Systems: Internals and Design Principles by Stallings (Pearson)
2. Operating Systems by J. Archer Harris (Author), Jyoti Singh (Author) (TMH)

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Case Study on a specific Operating System: highlighting its functions and key features.

Evaluation Method: Case study presentation, depth of understanding of operating system functions, and ability to articulate key concepts.

Unit 2: Activity: Comparison Poster on Scheduling Algorithms

Evaluation Method: Assessment of posters based on content accuracy, clarity of information, visual presentation, and ability to convey key insights.

Unit 3: Activity: Assignment on Dead Lock prevention techniques

Evaluation Method: Understanding, Completion and report.

Unit 4: Activity: Debate on various Memory allocation schemes

Evaluation Method: Debate arguments, ability to counter opposing viewpoints, logical reasoning, and presentation skills.

Unit 5: Activity: Comparative study of various disk scheduling algorithms using real world datasets

Evaluation Method: Analysis methodology, accuracy of results, and presentation of findings and conclusions.

III Semester
Course 8: Operating Systems
Credits -1

List of Experiments:

1. Illustrate the LINUX commands
 - a) pwd
 - b) mkdir
 - c) rmdir
 - d) grep
 - e) chmod
 - f) ls
 - g) rm
 - h) cp
2. Write a program to calculate average waiting time and turn around time of each process using the following CPU Scheduling algorithm for the given process schedules.
 - a) FCFS
 - b) SJF
 - c) Priority
 - d) Round Robin
3. Simulate MVT and MFT memory management techniques
4. Write a program for Bankers Algorithm for Dead Lock Avoidance
5. Implement Bankers Algorithm Dead Lock Prevention.
6. Write a program to simulate Producer-Consumer problem.
7. Simulate all Page replacement algorithms.
 - e) FIFO
 - f) LRU
 - g) LFU
 - h) Optimal
8. Simulate Paging Techniques of memory management
9. Simulate the following disk scheduling algorithms
 - a) FCFS
 - b) SSTF
 - c) SCAN
 - d) CSCAN

IV Semester
Course 9: Database Management Systems
Credits -3

Learning Objectives:

To familiarize with concepts of database design

Learning Outcomes: On successful completion of the course, students will be able to

1. Differentiate between database systems and file based systems
2. Design a database using ER model
3. Use relational model in database design
4. Use SQL commands for creating and manipulating data stored in databases.
5. Write PL/SQL programs to work with databases.

UNIT - I

Overview of Database Management System: Introduction to data, information, database, database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach, Various Data Models, Components of Database Management System, three schema architecture of data base, costs and risks of database approach.

UNIT - II

Entity-Relationship Model: Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, enhanced entity-relationship model (EER model), generalization and specialization, IS A relationship and attribute inheritance, multiple inheritance, constraints on specialization and generalization, advantages of ER modeling.

UNIT - III

Relational Model: Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, relational calculus, tuple relational calculus, domain relational Calculus (DRC), Functional dependencies and normal forms upto 3rd normal form.

UNIT - IV

Structured Query Language: Introduction, Commands in SQL, Data Types in SQL, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Join Operation, Set Operations, View, Sub Query.

UNIT - V

PL/SQL: Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements, Data Types, Operators Precedence, Control Structure, Steps to Create a PL/SQL, Program, Iterative Control, Procedure, Function, Database Triggers, Types of Triggers.

Text Books:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne (7th Edition) Wiley India Edition.

Reference Books

1. Database Management Systems by Raghu Ramakrishnan, McGrawhill
2. Principles of Database Systems by J. D. Ullman
3. Fundamentals of Database Systems by R. Elmasri and S. Navathe
4. SQL: The Ultimate Beginners Guide by Steve Tale.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Seminar Presentation on Database Management Systems

Evaluation Method: Depth of research, clarity of explanations, ability to address questions and engage the audience.

Unit 2: Activity: Case Study on EER model

Evaluation Method: Identification of inheritance relationships, effective use of generalization and specialization, and adherence to constraints.

Unit 3: Activity: Exercise on Normalization: Assign students a set of unnormalized tables and have them normalize the tables to third normal form

Evaluation Method: Normalized table designs, identification of functional dependencies, adherence to normalization rules, and elimination of anomalies.

Unit 4: Activity: Competition on SQL Query Writing

Evaluation Method: Query correctness, efficiency, proper use of SQL commands, ability to handle complex scenarios, and creativity in query formulation.

Unit 5: Activity: Peer Review of PL/SQL code

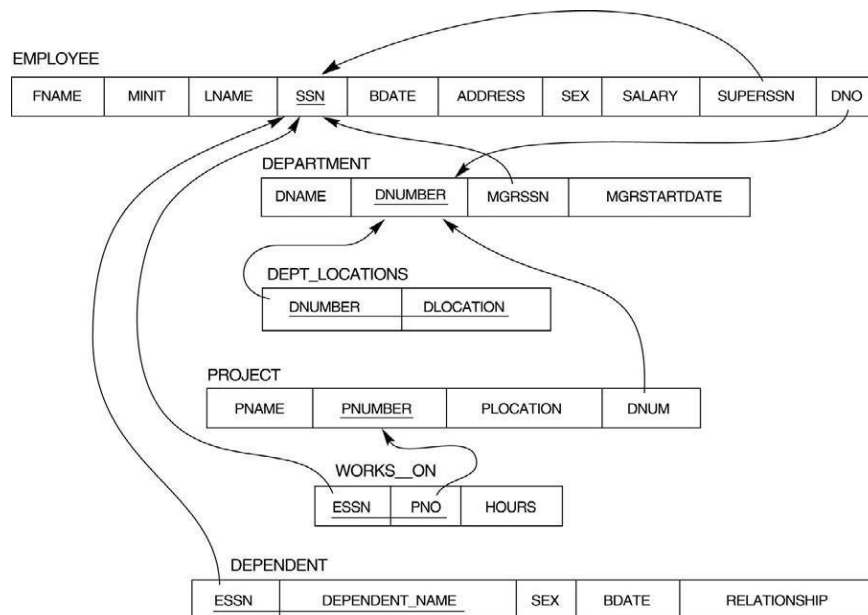
Evaluation Method: Peer evaluation of code quality, adherence to coding standards, proper use of language elements, and logic.

IV Semester
Course 9: Database Management Systems
Credits -1

List of Experiments:

1. Draw ER diagram for hospital administration
2. Creation of college database and establish relationships between tables
3. Relational database schema of a company is given in the following figure.

Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create above tables with relevant Primary Key, Foreign Key and other constraints
2. Populate the tables with data
3. Display all the details of all employees working in the company.
4. Display ssn, lname, fname, address of employees who work in department no 7.
5. Retrieve the Birthdate and Address of the employee whose name is 'Franklin T. Wong'
6. Retrieve the name and salary of every employee
7. Retrieve all distinct salary values
8. Retrieve all employee names whose address is in 'Bellaire'
9. Retrieve all employees who were born during the 1950s
10. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)

11. Retrieve the names of all employees who do not have supervisors
12. Retrieve SSN and department name for all employees
13. Retrieve the name and address of all employees who work for the 'Research' department
14. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.
15. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.
16. Retrieve all combinations of Employee Name and Department Name
17. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project.
18. Increase the salary of all employees working on the 'Product X' project by 15%. Retrieve employee name and increased salary of these employees.
19. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name.
20. Select the names of employees whose salary does not match with salary of any employee in department 10.
21. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
22. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
23. Find the sum of the salaries and number of employees of all employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
24. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
25. Delete all dependents of employee whose ssn is '123456789'.
26. Perform a query using alter command to drop/add field and a constraint in Employee table.

IV Semester
Course 10: Object Oriented Software Engineering
Credits -3

Course Objective:

To introduce Object-oriented software engineering (OOSE) - which is a popular technical approach to analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Understand and apply the fundamental principles of Object-Oriented Programming (OOP) concepts and Unified Modeling Language (UML) basics, in the development of software solutions.
2. Analyze and specify software requirements, develop use cases and scenarios, apply object-oriented analysis and design (OOAD) principles
3. Familiar with the concept of test-driven development (TDD) and its practical implementation
4. Analyze and Evaluate Software Maintenance and Evolution Strategies
5. Apply Advanced Object-Oriented Software Engineering Concepts

UNIT-I

Introduction to Object-Oriented Programming: Overview of software engineering, Introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, Introduction to software development process and software development life cycle (SDLC).

UNIT-II

Requirements Analysis and Design: Requirements analysis and specification, Use cases and scenarios, Object-oriented analysis and design (OOAD), Design patterns, UML modelling techniques (class diagrams, sequence diagrams, state machine diagrams, activity diagrams)

UNIT-III

Software Construction and Testing: Software construction basics, Object-oriented design principles, Object-oriented programming languages (Java, C++, Python), Software testing basics (unit testing, integration testing, system testing), Test-driven development (TDD)

UNIT-IV

Software Maintenance and Evolution: Software maintenance basics, refactoring techniques Software version control, Code review and inspection, Software evolution and reengineering

UNIT-V

Advanced Topics in Object-Oriented Software Engineering: Model-driven engineering (MDE), Aspect-oriented programming (AOP), Component-based software engineering (CBSE), Service-oriented architecture (SOA), Agile software development and Scrum methodologies.

Text Book(s)

1. An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd Edition, Craig Larman, Prentice-Hall.
2. Programming in Java by Sachin Malhotra, Oxford University Press

Reference Books

1. Requirements engineering: processes and techniques, G.Kotonya and, I.Sommerville, 1998, Wiley
2. Design Patterns, E.Gamma, R. Helm, R. Johnson, and J. Vlissides
3. The Unified Modeling Language Reference Manual, J. Rumbaugh, I.Jacobson and G. Booch, Addison Wesley

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Group Activity: Design and implement a small OOP project

Evaluation Method: Presentation evaluation rubric, Project evaluation based on OOP principles.

Unit 2: Activity: Use Case Scenario Presentation & Peer Activity: Review and provide feedback on each other's use case diagrams

Evaluation Method: Presentation evaluation rubric, Peer feedback assessment.

Unit 3: Activity: Poster Presentation: Illustrate TDD principles and benefits

Evaluation Method: Poster presentation evaluation

Unit 4: Activity: Peer Activity: Analyze and discuss different maintenance strategies

Evaluation Method: Peer discussion participation evaluation

Unit 5: Activity: Seminar on Design Patterns

Evaluation Method: Depth of research, clarity of explanations, ability to address questions and engage the audience.

IV Semester
Course 10: Object Oriented Software Engineering
Credits -1

Suggested Software Tools: StarUML/UMLGraph/Topcased/Umberollo/ArgoUML/ Eclipse IDE, Visual Paradigm for UML/Rational Software Architect/Any other Open Source Tool

List of Experiments:

Select domain of interest (e.g. College Management System) and identify multi-tier software application to work on (e.g. Online Fee Collection). Analyze, design and develop this application using OOSE approach:

1. Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart).
2. Understanding of System modeling: Data model i.e. ER – Diagram and draw the ER Diagram with generalization, specialization and aggregation of specified problem statement
3. Understanding of System modeling: Functional modeling: DFD level 0 i.e. ContextDiagram and draw it
4. Understanding of System modeling: Functional modeling: DFD level 1 and DFD level 2 and draw it.
5. Identify use cases and develop the use case model.
6. Identify the business activities and develop an UML Activity diagram.
7. Identify the conceptual classes and develop a domain model with UML Class diagram.
8. Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
9. Draw the state chart diagram.
10. Identify the user interface, domain objects, and technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
11. Implement the technical services layer.
12. Implement the domain objects layer.
13. Implement the user interface layer.
14. Draw component and deployment diagrams.

IV Semester
Course 11: Data Communication and Computer Networks
Credits -3

Course Objectives

To provide students with a comprehensive understanding of networking principles, protocols, and technologies, enabling them to design, analyze, and evaluate efficient and reliable network solutions.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand and apply network applications, hardware, software, and reference models for network communication.
2. Design and analyze data link layer protocols, multiple access protocols, and wireless LAN technologies.
3. Design routing algorithms, congestion control algorithms, and evaluate network layer protocols for internetworking.
4. Analyze transport service, transport protocols, and evaluate UDP and TCP in the internet.
5. Understand and evaluate application layer protocols, including DNS, email, WWW, and network management protocols.

UNIT-I

INTRODUCTION: Network applications, network hardware, network software, reference models: OSI, TCP/IP, Internet, Connection oriented network - X.25, frame relay.

THE PHYSICAL LAYER: Theoretical basis for communication, guided transmission media, wireless transmission, the public switched telephone networks, mobile telephone system.

UNIT-II

THE DATA LINK LAYER: Design issues, error detection and correction, elementary data link protocols, sliding window protocols, example data link protocols - HDLC, the data link layer on the internet.

THE MEDIUM ACCESS SUBLAYER: Channel allocations problem, multiple access protocols, Ethernet, Data Link Layer switching, Wireless LAN, Broadband Wireless, Bluetooth.

UNIT-III

THE NETWORK LAYER: Network layer design issues, routing algorithms, Congestion control algorithms, Internetworking, the network layer in the internet (IPv4 and IPv6), Quality of Service.

UNIT-IV

THE TRANSPORT LAYER: Transport service, elements of transport protocol, SimpleTransport Protocol, Internet transport layer protocols: UDP and TCP.

UNIT-V

THE APPLICATION LAYER: Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http.

APPLICATION LAYER PROTOCOLS: Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol, Telnet.

Text Book(s)

1. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India

Reference Books

2. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.
3. Kurose, Ross (2010), Computer Networking: A top down approach, Pearson Education, India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Hands-on exercises to configure network applications

Evaluation Method: Practical skills in configuring network applications, hardware, and software.

Unit 2: Activity: Protocol Design and Simulation using simulation tools like NS-3 or Cisco Packet Tracer.

Evaluation Method: Students' ability to design and simulate data link layer protocols and multiple access protocols

Unit 3: Activity: Guest Lectures and Workshops on routing algorithms, congestion control, and network layer protocols.

Evaluation Method: Students' participation and understanding demonstrated in guest lectures and workshop

Unit 4: Activity: Network Monitoring and Traffic Analysis using tools like Wireshark

Evaluation Method: Understanding of transport protocols through their analysis of network traffic and identification of UDP and TCP behavior

Unit 5: Activity: Group Projects on Network Application Development

Evaluation Method: Group Project Presentations

Semester - IV
Course 11: Data Communication and Computer Networks
Credits -1

List of Experiments:

1. Understanding various network tools in Windows and Linux
2. Study different types of Network devices and Cables
3. Building a Local Area Network
4. Concept of Network IP Address
5. Introduction to Network Simulator – Packet Tracer (PT)
6. Configuration of a Router using Packet Tracer
7. Implementation of a Network using Packet Tracer
8. Implementation of Static Routing using Packet Tracer
9. Implementation of RIP using Packet Tracer
10. Implementation of OSPF using Packet Tracer
11. Implement DNS using packet tracer
12. Implementation of a VLAN using Packet Tracer

ANDHRA KESARI UNIVERSITY-ONGOLE, PRAKASAM DISTRICT
Single Major Programme from the Year 2023-24 Onwards
Programme-B.Sc. Honours Computer Science - Question Paper model,
Second Year-Semester-III & IV

Time: 3 Hours

Total Marks: 75

PART –A

Answer any Five of the following

5X5=25 Marks

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

PART –B

Answer the following

5x10=50 Marks

- 11.
 - 12.
 - 13.
 - 14.
 - 15.
 - 16.
 - 17.
 - 18.
 - 19.
 - 20.
-

