ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory body of the Government of Andhra Pradesh)

3rd,4th and 5th floors, Neeladri Towers,Sri Ram Nagar,6th Battalion Road,
Atmakur(V),Mangalagiri(M), Guntur-522 503, Andhra Pradesh
Web:www.apsche.org **Email**: acapsche@gmail.com

REVISED SYLLABUS OF B.Sc. (COMPUTER SCIENCE/ INFORMATION TECHNOLOGY) UNDER CBCS FRAMEWORK WITH EFFECT FROM 2020-2021

PROGRAMME: THREE-YEAR B.Sc.

(B.Sc. Computer Science/ Information Technology (IT))

(With Learning Outcomes, Unit-wise Syllabus, References, Co-curricular Activities & Model Q.P.)

For Fifteen Courses of 1, 2, 3 & 4 Semesters)

(To be Implemented from 2020-21 Academic Year)

Structure of Computer Science /Information Technology (IT)

Programme: B.Sc. with Computer Science as one of the Core Subjects.
Discipline: Computer Science

Year	Semester	Paper Code	Subject	Hrs. per Week	Credits	IA	ES	Total
First Year	I	C1	Problem Solving in C	4	3	25	75	100
	I	C1-P	Problem Solving in C Lab	2	2		50	50
	II	C2	Data Structures using C	4	3	25	75	100
	II	C2-P	Data Structures using C Lab	2	2		50	50
Second Year	III	C3	Database Management System	4	3	25	75	100
	III	C3-P	Database Management System Lab	2	2		50	50
	IV	C4	Object Oriented Programming using Java	4	3	25	75	100
	IV	C4-P	Object Oriented Programming using Java Lab	2	2		50	50
	IV	C5	Operating Systems	4	3	25	75	100
	IV	C5-P	Operating Systems Lab using C/Java	2	2		50	50

PROBLEM SOLVING IN C

Semester	Course Code	Course Title	Hours	Credits
I	C1	PROBLEM SOLVING IN C	60	3

Objectives:

This course aims to provide exposure to problem-solving through programming. It introduces the concepts of the C Programming language.

Course Learning Outcomes:

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

1. Understand the evolution and functionality of a Digital Computer.
2. Apply logical skills to analyse a given problem
3. Develop an algorithm for solving a given problem.
4. Understand 'C' language constructs like Iterative statements, Array processing, Pointers, etc.
5. Apply 'C' language constructs to the algorithms towrite a 'C' language program.

UNIT I

General Fundamentals: Introduction to computers: Block diagram of a computer, characteristics and limitations of computers, applications of computers, types of computers, computer generations.

Introduction to Algorithms and Programming Languages: Algorithm – Key features of Algorithms, Flow Charts, Programming Languages – Generations of Programming Languages – Structured Programming Language- Design and Implementation of Correct, Efficient and Maintainable Programs.

UNIT II

Introduction to C: Introduction – Structure of C Program – Writing the first C Program – File used in C Program – Compiling and Executing C Programs – Using Comments –

Keywords – Identifiers – Basic Data Types in C – Variables – Constants – I/O Statements in C- Operators in C- Programming Examples.

Decision Control and Looping Statements: Introduction to Decision Control Statements– Conditional Branching Statements – Iterative Statements – Nested Loops – Break and Continue Statement – Goto Statement

UNIT III

Arrays: Introduction – Declaration of Arrays – Accessing elements of the Array – Storing Values in Array– Operations on Arrays – one dimensional, two dimensional and multi dimensional arrays, character handling and strings.

UNIT IV

Functions: Introduction – using functions – Function declaration/ prototype – Function definition – function call – return statement – Passing parameters – Scope of variables – Storage Classes – Recursive functions.

Structure, Union, and Enumerated Data Types: Introduction – Nested Structures – Arrays of Structures – Structures and Functions– Union – Arrays of Unions Variables – Unions inside Structures – Enumerated Data Types.

UNIT V

Pointers: Understanding Computer Memory – Introduction to Pointers – declaring Pointer Variables – Pointer Expressions and Pointer Arithmetic – Null Pointers - Passing Arguments to Functions using Pointer – Pointer and Arrays – Memory Allocation in C Programs – Memory Usage – Dynamic Memory Allocation – Drawbacks of Pointers

Files: Introduction to Files – Using Files in C – Reading Data from Files – Writing Data to Files – Detecting the End-of-file – Error Handling during File Operations – Accepting Command Line Arguments.

BOOKS

1. E Balagurusamy – Programming in ANSIC – Tata McGraw-Hill publications.
2. Brain W Kernighan and Dennis M Ritchie - The 'C' Programming language" - Pearson publications.
3. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publications.
4. YashavantKanetkar - Let Us 'C' – BPB Publications.

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Problem-solving exercises,
4. Practical assignments and laboratory reports,

5. Observation of practical skills,
6. Individual and group project reports like “Creating Text Editor in C”.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
I	**C1-P**	**PROBLEM SOLVING IN C LAB**	**30**	**2**

Problem solving in C LAB

1. Write a program to check whether the given number is Armstrong or not.
2. Write a program to find the sum of individual digits of a positive integer.
3. Write a program to generate the first n terms of the Fibonacci sequence.
4. Write a program to find both the largest and smallest number in a list of integer values
5. Write a program to demonstrate refection of parameters in swapping of two integer values using **Call by Value**&**Call by Address**
6. Write a program that uses functions to add two matrices.
7. Write a program to calculate factorial of given integer value using recursive functions
8. Write a program for multiplication of twoN X N matrices.
9. Write a program to perform various string operations.
10. Write a program to search an element in a given list of values.
11. Write a program to sort a given list of integers in ascending order.
12. Write a program to calculate the salaries of all employees using ***Employee (ID, Name, Designation, Basic Pay, DA, HRA, Gross Salary, Deduction, Net Salary)*** structure.
    a. DA is 30 % of Basic Pay
    b. HRA is 15% of Basic Pay
    c. Deduction is 10% of (Basic Pay + DA)
    d. Gross Salary = Basic Pay + DA+ HRA
    e. Net Salary = Gross Salary - Deduction
13. Write a program to illustrate pointer arithmetic.

14. Write a program to read the data character by character from a file.
15. Write a program to create***Book (ISBN,Title, Author, Price, Pages, Publisher)***structureand store book details in a file and perform the following operations
    a. Add book details
    b. Search a book details for a given ISBN and display book details, if available
    c. Update a book details using ISBN
    d. Delete book details for a given ISBN and display list of remaining Books

DATA STRUCTURES USING C

Semester	Course Code	Course Title	Hours	Credits
II	C2	DATA STRUCTURES USING C	60	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 Learning Outcomes:

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

1. Understand available Data Structures for data storage and processing.
2. ComprehendData Structure and their real-time applications - Stack, Queue, Linked List, Trees and Graph
3. Choose a suitable Data Structures for an application
4. Develop ability to implement different Sorting and Search methods
5. Have knowledge onData Structures basic operations like insert, delete, search,update and traversal
6. Design and develop programs using various data structures
7. Implement the applications of algorithms for sorting, pattern matching etc

UNIT – I:

Introduction to Data Structures: Introduction to the Theory of Data Structures, Data Representation, Abstract Data Types, Data Types, Primitive Data Types, Data Structure and Structured Type, Atomic Type, Difference between Abstract Data Types, Data Types, and Data Structures, Refinement Stages

Principles of Programming and Analysis of Algorithms: Software Engineering, Program Design, Algorithms, Different Approaches to Designing an Algorithm, Complexity, Big ‘O’ Notation, Algorithm Analysis, Structured Approach to Programming, Recursion, Tips and Techniques for Writing Programs in ‘C’

UNIT – II:

Arrays: Introduction to Linear and Non- Linear Data Structures, One- Dimensional Arrays, Array Operations, Two- Dimensional arrays, Multidimensional Arrays, Pointers and Arrays, an Overview of Pointers

Linked Lists: Introduction to Lists and Linked Lists, Dynamic Memory Allocation, Basic Linked List Operations, Doubly Linked List, Circular Linked List, Atomic Linked List, Linked List in Arrays, Linked List versus Arrays

UNIT – III:

Stacks: Introduction to Stacks, Stack as an Abstract Data Type, Representation of Stacks through Arrays, Representation of Stacks through Linked Lists, Applications of Stacks, Stacks and Recursion

Queues: Introduction, Queue as an Abstract data Type, Representation of Queues, Circular Queues, Double Ended Queues- Deques, Priority Queues, Application of Queues

UNIT – IV:

Binary Trees: Introduction to 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, Counting Number of Binary Trees, Applications of Binary Tree

UNIT – V:

Searching and sorting: Sorting – An Introduction, Bubble Sort, Insertion Sort, Merge Sort, Searching – An Introduction, Linear or Sequential Search, Binary Search, Indexed Sequential Search

Graphs: Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs, Spanning Trees, Shortest Path, Application of Graphs.

BOOKS:

1. “Data Structures using C”, ISRD group Second Edition, TMH

2. “Data Structures through C”, YashavantKanetkar, BPB Publications
3. “Data Structures Using C” Balagurusamy E. TMH

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. **Measurable**

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. **General**

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
II	**C2-P**	**DATA STRUCTURES USING C LAB**	**30**	**2**

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 begging of an array
    b. Insert an element at given index of array
    c. Update a element using a values and index
    d. Delete an existing element
2. Write a program using stacks to convert a given
    a. postfix expression to prefix
    b. prefix expression to postfix
    c. infix expression to postfix
3. Write Programs to implement the Stack operations using an array
4. Write Programs to implement the Stack operations using Liked List.
5. Write Programs to implement the Queue operations using an array.
6. Write Programs to implement the Queue operations using Liked List.
7. Write a program for arithmetic expression evaluation.
8. Write a program for Binary Search Tree Traversals
9. Write a program to implement dequeue using a doubly linked list.
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
12. Write a program for polynomial addition using single linked list
13. Write a program to find out shortest path between given Source Node and Destination Node in a given graph using Dijkstrar's algorithm.

14. Write a program to implement Depth First Search graph traversals algorithm
15. Write a program to implement Breadth First Search graph traversals algorithm

DATABASE MANAGEMENT SYSTEMS

Semester	Course Code	Course Title	Hours	Credits
III	C3	DATABASE MANAGEMENT SYSTEMS	60	3

Course Objective:

The objective of the course is to introduce the design and development of databases with special emphasis on relational databases.

Course Learning Outcomes:

On completing the subject, students will be able to:

1. Gain knowledge of Database and DBMS.
2. Understand the fundamental concepts of DBMS with special emphasis on relational data model.
3. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database
4. Model databaseusing ER Diagrams and design database schemas based on the model.
5. Create a small database using SQL.
6. Store, Retrieve data in database.

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 modelling.

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, History of SQL Standard, 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.

BOOKS:

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

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

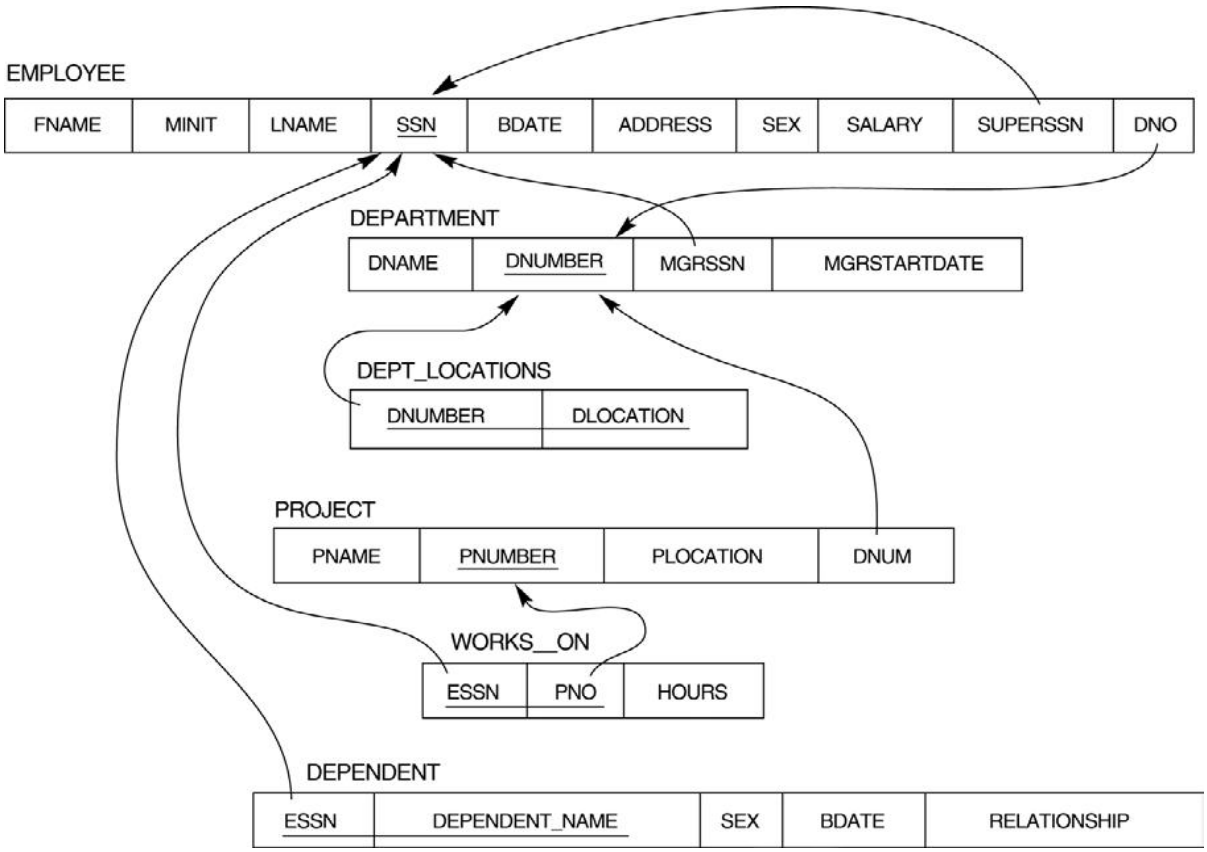
1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Practical assignments and laboratory reports,
4. Observation of practical skills,

5. Individual and group project reports like Create your college database for placement purpose.
6. Efficient delivery using seminar presentations,
7. Viva voce interviews.
8. Computerized adaptive testing, literature surveys and evaluations,
9. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
III	**C3-P**	**DATABASE MANAGEMENT SYSTEMS LAB**	**30**	**2**

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 'ProductX' 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.

OBJECT ORIENTATED PROGRAMMING THROUGH JAVA

Semester	Course Code	Course Title	Hours	Credits
IV	**C4**	**OBJECT ORIENTATED PROGRAMMING THROUGH JAVA**	**60**	**3**

Objectives:

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

Course Learning Outcomes: At the end of this course student will:

1. Understand the benefits of a well-structured program
2. Understand different computer programming paradigms
3. Understand underlying principles of Object-Oriented Programming in Java
4. Develop problem-solving and programming skills using OOP concepts
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java

UNIT – I

Introduction to Java: Features of Java, The Java virtual Machine, Parts of Java

Naming Conventions and Data Types: Naming Conventions in Java, Data Types in Java, Literals

Operators in Java: Operators, Priority of Operators

Control Statements in Java: if... else Statement, do... while Statement, while Loop, for Loop, switch Statement, break Statement, continue Statement, return Statement

Input and Output: Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format()

Arrays: Types of Arrays, Three Dimensional Arrays (3D array), arrayname.length, Command Line Arguments

UNIT – II

Strings: Creating Strings, String Class Methods, String Comparison, Immutability of Strings

Introduction to OOPs: Problems in Procedure Oriented Approach, Features of Object-Oriented Programming System (OOPS)

Classes and Objects: Object Creation, Initializing the Instance Variables, Access Specifiers, Constructors

Methods in Java: Method Header or Method Prototype, Method Body, Understanding Methods, Static Methods, Static Block, The keyword 'this', Instance Methods, Passing Primitive Data Types to Methods, Passing Objects to Methods, Passing Arrays to Methods, Recursion, Factory Methods

Inheritance: Inheritance, The keyword 'super', The Protected Specifier, Types of Inheritance

UNIT – III

Polymorphism: Polymorphism with Variables, Polymorphism using Methods, Polymorphism with Static Methods, Polymorphism with Private Methods, Polymorphism with Final Methods, final Class

Type Casting: Types of Data Types, Casting Primitive Data Types, Casting Referenced Data Types, The Object Class

Abstract Classes: Abstract Method and Abstract Class

Interfaces: Interface, Multiple Inheritance using Interfaces

Packages: Package, Different Types of Packages, The JAR Files, Interfaces in a Package, Creating Sub Package in a Package, Access Specifiers in Java, Creating API Document

Exception Handling: Errors in Java Program, Exceptions, throws Clause, throw Clause, Types of Exceptions, Re – throwing an Exception

UNIT – IV

Streams: Stream, Creating a File using FileOutputStream, Reading Data from a File uingFileInputStream, Creating a File using FileWriter, Reading a File using FileReader, Zipping and Unzipping Files, Serialization of Objects, Counting Number of Characters in a File, File Copy, File Class

Threads: Single Tasking, Multi Tasking, Uses of Threads, Creating a Thread and Running it, Terminating the Thread, Single Tasking Using a Thread, Multi Tasking Using Threads, Multiple Threads Acting on Single Object, Thread Class Methods, Deadlock of Threads,

Thread Communication, Thread Priorities, thread Group, Daemon Threads, Applications of Threads, Thread Life Cycle

UNIT – V

Applets: Creating an Applet, Uses of Applets, <APPLET> tag, A Simple Applet, An Applet with Swing Components, Animation in Applets, A Simple Game with an Applet, Applet Parameters

Java Database Connectivity: Database Servers, Database Clients, JDBC (Java Database Connectivity), Working with Oracle Database, Working with MySQL Database, Stages in a JDBC Program, Registering the Driver, Connecting to a Database, Preparing SQL Statements, Using jdbc–odbc Bridge Driver to Connect to Oracle Database, Retrieving Data from MySQL Database, Retrieving Data from MS Access Database, Stored Procedures and CallableStatements, Types of Result Sets

BOOKS:

1. Core Java: An Integrated Approach, Authored by Dr. R. Nageswara Rao &Kogent Learning Solutions Inc.
2. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.
3. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TMH.
4. Deitel&Deitel. Java TM: How to Program, PHI (2007)

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,

4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
IV	**C4-P**	**OBJECT ORIENTATED PROGRAMMING THROUGH JAVA LAB**	**30**	**2**

1. Write a program to read ***Student Name, Reg.No, Marks[5]*** and calculate ***Total***, ***Percentage, Result***. Display all the details of students
2. Write a program to perform the following String Operations
    - **a.** Read a string
    - **b.** Find out whether there is a given substring or not
    - **c.** Compare existing string by another string and display status
    - **d.** Replace existing string character with another character
    - **e.** Count number of works in a string
3. Java program to implements Addition and Multiplication of two N X N matrices.
4. Java program to demonstrate the use of Constructor.
5. Calculate area of the following shapes using method overloading.
    - **a.** Triangle
    - **b.** Rectangle
    - **c.** Circle
    - **d.** Square
6. Implement inheritance between ***Person (Aadhar, Surname, Name, DOB, and Age)*** and ***Student (Admission Number, College, Course, Year)***classes where ReadData(), DisplayData() are overriding methods.
7. Java program for implementing Interfaces
8. Java program on Multiple Inheritance.
9. Java program for to display ***Serial Number from 1 to N*** by creating two Threads
10. Java program to demonstrate the following exception handlings
    a. Divided by Zero
    b. Array Index Out of Bound
    c. File Not Found
    d. Arithmetic Exception
    e. User Defined Exception

11. Create an Applet to display different shapes such as Circle, Oval, Rectangle, Square and Triangle.

12. Write a program to create ***Book (ISBN,Title, Author, Price, Pages, Publisher)***structure and store book details in a file and perform the following operations
    a. Add book details
    b. Search a book details for a given ISBN and display book details, if available
    c. Update a book details using ISBN
    d. Delete book details for a given ISBN and display list of remaining Books

OPERATING SYSTEMS

Semester	Course Code	Course Title	Hours	Credits
IV	C5	OPERATING SYSTEMS	60	2

Objectives:

This course aims to introduce the structure and organization of a file system. It emphasizes various functions of an operating system like memory management, process management, device management, etc.

Course Learning Outcomes:

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

1. Know Computer system resources and the role of operating system in resource management with algorithms
2. Understand Operating System Architectural design and its services.
3. Gain knowledge of various types of operating systems including Unix and Android.
4. Understand various process management concepts including scheduling, synchronization, and deadlocks.
5. Have a basic knowledge about multithreading.
6. Comprehenddifferent approaches for memory management.
7. Understand and identify potential threats to operating systems and the security features design to guard against them.
8. Specify objectives of modern operating systems and describe how operating systems have evolved over time.
9. Describe the functions of a contemporary operating system

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, ProcessAbstraction, ProcessHierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling, Non-Preemptive and Preemptive Scheduling Algorithms.

UNIT III

Process Management: Deadlock, Deadlock Characterization, Necessary andSufficient Conditions forDeadlock, 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; MemoryAllocation Strategies– Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT V

File and I/O Management, OSsecurity : DirectoryStructure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Security Policy Mechanism, Protection, Authentication and Internal Access Authorization

Introduction to Android Operating System, Android Development Framework, Android Application Architecture, Android Process Management and File System, Small Application Development using Android Development Framework.

REFERENCE BOOKS:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne (7thEdition) Wiley India Edition.
2. Operating Systems: Internals and Design Principles by Stallings (Pearson)
3. Operating Systems by J. Archer Harris (Author), Jyoti Singh (Author) (TMH)
4. Online Resources for UNIT V

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.

7. Efficient delivery using seminar presentations,
8. Viva-Voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
IV	**C-5**	**OPERATING SYSTEMS LAB USING C/Java**	**30**	**2**

1. Write a program to implement Round Robin CPU Scheduling algorithm
2. Simulate SJF CPU Scheduling algorithm
3. Write a program the FCFS CPU Scheduling algorithm
4. Write a program to Priority CPU Scheduling algorithm
5. Simulate Sequential file allocation strategies
6. Simulate Indexed file allocation strategies
7. Simulate Linked file allocation strategies
8. Simulate MVT and MFT memory management techniques
9. Simulate Single level directory File organization techniques
10. Simulate Two level File organization techniques
11. Simulate Hierarchical File organization techniques
12. Write a program for Bankers Algorithm for Dead Lock Avoidance
13. Implement Bankers Algorithm Dead Lock Prevention.
14. Simulate all Page replacement algorithms.
    a) FIFO
    b) LRU
    c) LFU
15. Simulate Paging Techniques of memory management

SUBJECT EXPERTS

Dr.M.Ussenaiah
Dept of Computer Science,
Vikrama Simhapuri University

Dr.A.Kavitha,
Govt. Degree College,
Repalle

SYLLABUS VETTED BY

Dr.Gangadhar,
Dept of Computer Science
Acgharya Nagarjuna University,
Nagarjuna Nagar