

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 to write 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. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publications.
3. Yashavant Kanetkar - Let Us 'C' – BPB Publications.

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 refaction 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 two N 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