Course Title: **Programming Technique using C**
Course no: **CA104CS** Credit hours: **3**
Year / Semester: I/**I** Full Marks: 60+20+20
Nature of course: Theory + Lab Pass Marks: 24+08+08

Course Description

This course is designed to introduce students to the fundamental concepts of computer programming using the C language. It emphasizes problem-solving through structured programming, guiding students from algorithm design and flowchart creation to the development of complete C programs. The syllabus covers essential topics such as variables, data types, operators, control structures, functions, arrays, pointers, strings, structures, unions, and file handling, along with an introduction to graphics programming. Through a blend of theoretical instruction and hands-on lab exercises, students gain practical experience in writing, testing, and debugging C programs.

Course Objectives:

1. Develop the ability in students to analyze real-life problems and design algorithmic solutions using structured programming concepts.
2. Enable student's ability to translate algorithms into efficient and correct C programs.
3. Introduce students to the syntax, semantics, and structure of the C programming language.
4. Make Familiarization to students with control structures, functions, arrays, pointers, and file handling in C.
5. Encourage the student to problem-solving skills through practical programming exercises and projects.
6. Prepare students to build simple applications that include input, processing, output, and data storage using C.

Contents

Unit 1: Introduction to Programming Concept **[4 Hrs.]**

- 1.1 Introduction of Programming Language
- 1.2 Assembler, Compiler and Interpreter
- 1.3 Syntax and Semantics
- 1.4 Programming Design Tools
  - 1.4.1 Algorithm
  - 1.4.2 Flow chart
  - 1.4.3 Pseudo codes
- 1.5 Features of good program

Lab Works

- Designing algorithm and draw flow chart for sequence, decision making and repetition concept of general programming using any CASE tool.

Unit 2: Introduction to C **[6 Hrs.]**

- 2.1 History of C Program
- 2.2 Basic Structure of C Program
- 2.3 Character set, Token and Comments
- 2.4 Variables and Constants
- 2.5 Data Types
- 2.6 Type Conversion
- 2.7 Input and Output
- 2.8 Operators
  - 2.8.1 Arithmetic

2.8.2 Relational
2.8.3 Logical
2.8.4 Increment/Decrement
2.8.5 Assignment
2.8.6 Bitwise
2.8.7 Ternary/ Conditional (:?)
2.8.8 Comma

Lab Works

- Developing basic structure of C program
- Declaring and assigning variables and constants.
- Applying input and output build in function
- Using arithmetic operators
- Giving demo of type conversion

Unit 3: Control Structure **[7 Hrs.]**

1.1 Selective Structure
1.1.1 If statement
1.1.2 If-else statement
1.1.3 Nested if-else statement
1.1.4 Switch statement
1.1.5 Conditional operator (:?)

1.2 Looping structure:
1.2.1 While Loop
1.2.2 Do-while loop,
1.2.3 For loop
1.2.4 Nested Loops

3.2 Loop interrupts (jump, break, exit)

Practical Works

- User if, if-else and switch statement
- Use while, do-while, for and nested loop concept.

Unit 4: Function **[6 Hrs.]**

4.1 Function Concept
4.2 Function prototype, call and definition
4.3 Different ways of using function
4.4 Call by value, call by reference
4.5 Recursion

Practical Works

- Prototype, call and define function
- Pass the different parameter methods
- Use call by reference methods using function.
- Create a recursion function.

Unit 5: Array, Pointer and String **[7 Hrs.]**

5.1 Concept of array
5.2 Array declare, access and initialization.
5.3 Multi-dimensional array
5.4 Concept of Pointer
5.5 Pointer address, dereference, declaration, assignment, initialization
5.6 Pointer Arithmetic
5.7 Array and Pointer
5.8 String
5.9 String functions in C
5.10 Pointer and String

Practical Works

- Array declares, define, initialize.
- Creating a single or multi-dimensional array.

- Using pointer and demo for arithmetic function.
- Using different string function in program.

Unit 6: Structure and Union **[7 Hrs.]**

6.1 Concept of Structure
6.2 Initializing, accessing member of structure
6.3 Array of structure
6.4 Pointer to structure
6.5 Union
6.6 Different between union and structure

Practical Works

- Creating structure data types with application of loop.
- Creating union data types.

Unit 7: Memory Management **[6 Hrs.]**

7.1 Concept of Memory Management
7.2 Dynamic Memory Allocation (DMA)
7.3 Common Memory Management Issues
7.4 Pointer and Memory Relationship

Practical works

- Creating dynamic arrays using pointers
- Allocating memory for strings and structures dynamically
- Implementing simple memory management examples

Unit 8: Input output and File Handling **[5 Hrs.]**

7.1 Concept of File handling
7.2 File Access methods
7.3 Functions of file handling: fopen(), fclose(), fflush(), freopen()
7.4 Formatted input out
7.5 Character input output
7.6 Direct input output
7.7 Random file access
7.8 Error handling
7.9 File operation

Practical Works

- Creating file handling application for open, read, write and appends.
- Handling the random access files.
- Applying the text formatting function.

Micro Project

Preparation of simple application using C with the feature of input, process, output and store in external file.

TEXT BOOKS:

- Kanetkar, Y. P. (2008). *Let us C*8th Ed, New Delhi, BPB Publication (Unit 1 -8)
- Balagurusamy, E. (2007). Programming in ANSI C. New Delhi, India: Tata McGraw-Hill.

References materials:

- Raman, R. (1984). Computer programming in C, New Delhi, PHI. India
- Carlo Ghezi, Mehdi Jazayeri, "Programming Language Concepts", John Wiley and Sons
- B.S. Gottfried(2001), Schaum's Outline Series for Programming with C, Second Edition, Tata McGraw Hill Publishing Company, New Delhi