

C Programming

Course Title: C Programming

Course No: BIT102

Nature of the Course: Theory + Lab

Semester: I

Full Marks: 60 + 20 + 20

Pass Marks: 24 + 8 + 8

Credit Hrs: 3

Course Description:

This course familiarizes students with basic principles of programming. It introduces structured programming paradigm using a high-level language called C. It covers the concept of problem solving techniques, program design, and basic elements of C along with the detailed concept of operators, statements, arrays, functions, pointers, structures and file handling.

Course Objectives:

The main objective of this course is to acquaint the students with good program design through structured programming paradigm for developing programs for specific tasks using C Programming Language as well as to present the syntax and semantics of the “C” language.

Course Contents:

Unit I: Introduction (3 Hrs.)

History and advantage of C; Problems analysis, algorithm and flowchart; Structure of a C Program; Writing, compiling, debugging, executing and testing a C program in windows and Unix/Unix like environment.

Unit 2: Elements of C (3 Hrs.)

C Tokens; C Character Sets; Data types, constants and variables; Expression, statements and comments; Escape sequences and Delimiters.

Unit 3: Input/ Output function (2 Hrs.)

Conversion Specifiers; I/O functions; Formatted I/O

Unit 4: Operators and Expression (4 Hrs.)

Arithmetic operators; Relational operators; Logical operators; Assignment Operators; Type conversion in assignment; Increment and decrement operators; Ternary operator; Bitwise operator; Other Operators (comma, sizeof); Expression evaluation; Operator Precedence and Associativity.

Unit 5: Control Structures (8 Hrs.)

Introduction to selection and iterative statements; GOTO and labels; Selection statements: if, if...else, if...else...if ladder, nested if, switch case; Conditional operator; Iterative statement; For loop, While Loop, Do while Loop, Nested Loop; The odd loop; Controlling the loop execution-break and continue

Unit 6: Arrays and Strings (5 Hrs.)

Introduction to Arrays; Initializing Arrays; The meaning of array indexing; One dimensional and Multi-dimensional Arrays; String and Basic functions dedicated to string manipulation

Unit 7: Functions (6 Hrs.)

Introduction and types of functions; Declaring, Defining and calling functions; Arguments and Return Statement; Recursive functions; Function call by value and reference; Variables scope, local variables and function parameters; Arrays as function parameter; Void as a parameter; Parameterizing the main function; External function and variables; Header files; Static variables; Register variables

Unit 8: The C Processor (2 Hrs.)

Features of C processor; Macro Expansion; Macros with Arguments; Macros versus Functions; File Inclusion; Conditional Compilation; #if and #elif Directives; #undef Directive; #pragma Directive; The Build Process; Preprocessing; Compiling; Assembling; Linking; Loading

Unit 9: Pointers (5 Hrs.)

Introduction of Pointers, declaration and initialization of pointer variables; An address, a reference, a dereference and the sizeof operator; Pointer to nothing (NULL); Pointer assignment; Pointer Arithmetic; Pointer as argument and Pointer as return values; Pointers VS. arrays; Dynamic memory allocation

Unit 10: Structure and Unions (5 Hrs.)

Definition of Structure; Array of structures; Passing structure and array structure to function; Pointers to structures and arrays of structures; Self-referential structures; Typedef; Table Lookup; Unions

Unit 11: File Handling (2 Hrs.)

Files vs. streams; Header files needed for stream operations; Opening and closing a stream, open modes, errno variable; Reading and writing to/from a stream; Predefined streams; stdin, stdout and stderr; Stream manipulation: fgetc(), fputc(), fgets() and fputs() functions; Raw input/output: fread() and fwrite() functions; Random access to files

Laboratory Works:

Laboratory work emphasizes the verification of programming concepts learned in class. Therefore, each unit should include sufficient practical lab exercise.

Text / Reference Books:

1. Let us C, Yashvant P. Kanetkar
2. Brian Kernighan and Dennis Ritchie, The C Programming Language
3. Byron Gottfried, Programming with C, McGraw Hill Education