

# COMPUTER PROGRAMMING

ENCT 101

**Lecture : 3**

**Tutorial : 1**

**Practical : 3**

**Year : I**

**Part : I**

## **Course Objectives:**

The primary goal of this course is to provide students with a solid foundation in the principles of programming and to impart practical skills in the C programming language. This course ensures that students comprehend the fundamental concepts of variables, data types, control structures, and functions within the context of C. Advanced topics such as pointers, structures, file handling and the standard C library are explored to broaden students' programming capabilities. Also, through project-based assessments and evaluations, students apply their knowledge to real-world scenarios, fostering creativity and project development skills.

### **1 Introduction to Computer Programming**

**(3 hours)**

- 1.1 Definition of a computer program and programming language
- 1.2 Types and generations of programming languages
- 1.3 Problem-solving using a computer
  - 1.3.1 Problem analysis
  - 1.3.2 Algorithm and flowchart
  - 1.3.3 Programming
  - 1.3.4 Compilation, linking and execution
  - 1.3.5 Debugging and testing
  - 1.3.6 Documentation

### **2 Overview of C Programming**

**(3 hours)**

- 2.1 Introduction to C programming
- 2.2 History and importance of C
- 2.3 C headers and library functions
- 2.4 Basic structure of a C program
- 2.5 Preprocessor directives
- 2.6 Tokens in C (Character set, keywords and identifiers)
- 2.7 Type casting (Implicit and explicit)
- 2.8 Data types, variables and constants
- 2.9 Compiler and IDE for C programming

### **3 Operators and Expressions (4 hours)**

- 3.1 Introduction to operators and expressions
- 3.2 Arithmetic, relational and logical operators
- 3.3 Assignment, increment and decrement operators
- 3.4 Conditional, bitwise and special operators
- 3.5 Comma operator, size of operator
- 3.6 Evaluation and type conversion in expressions
- 3.7 Operator precedence and associativity

### **4 Input and Output (3 hours)**

- 4.1 Introduction to data I/O in C
- 4.2 Unformatted I/O
  - 4.2.1 Character I/O
  - 4.2.2 String I/O
- 4.3 Formatted I/O
  - 4.3.1 Control string (Flags, field width, precision, and specifier)
  - 4.3.2 Formatted I/O (scanf(), printf())

### **5 Control Structures (8 hours)**

- 5.1 Introduction to simple and compound statement
- 5.2 Sequential statement
- 5.3 Branching statement
  - 5.3.1 Simple if statement
  - 5.3.2 if-else statement
  - 5.3.3 Nested if-else statement
  - 5.3.4 else-if ladder
  - 5.3.5 switch statement
  - 5.3.6 go to statement
- 5.4 Looping statement
  - 5.4.1 for loop
  - 5.4.2 while loop
  - 5.4.3 do while
  - 5.4.4 Nested loop
- 5.5 Loop interruption
  - 5.5.1 break
  - 5.5.2 continue

### **6 Array and Pointer (7 hours)**

- 6.1 Introduction to an array
- 6.2 One-dimensional array
- 6.3 Two-dimensional array

- 6.4 Multidimensional array
- 6.5 Introduction to string
- 6.6 String handling functions
- 6.7 Definition of a pointer
- 6.8 Pointer declaration
- 6.9 Pointer arithmetic
- 6.10 Relationship between pointer and arrays

## **7 User-defined Functions**

**(6 hours)**

- 7.1 Introduction to function
- 7.2 Advantages of function
- 7.3 Elements of user-defined function
  - 7.3.1 Function definition
  - 7.3.2 Function prototype
  - 7.3.3 Function parameters
- 7.4 Storage class
- 7.5 Scope rules
- 7.6 Category of functions
  - 7.6.1 Functions with no arguments and no return values
  - 7.6.2 Functions with arguments and no return values
  - 7.6.3 Functions with arguments and return values
  - 7.6.4 Functions with no arguments and return values
- 7.7 Recursive functions
- 7.8 Function call by values and reference
- 7.9 Passing array and string to function

## **8 Structures**

**(5 hours)**

- 8.1 Defining a structure
- 8.2 Declaring and accessing structure elements
- 8.3 Initializing structure
- 8.4 Array of structure
- 8.5 Array as member to structure
- 8.6 Pointer as member to structure
- 8.7 Structure as a member to structure
- 8.8 Passing and returning structures to/from function

## **9 File Management**

**(4 hours)**

- 9.1 Introduction
- 9.2 Binary and text file in C
- 9.3 File opening modes
- 9.4 Defining, opening and closing file

- 9.5 Input-output operations on files
  - 9.5.1 Character I/O (fputc(), fgetc())
  - 9.5.2 String I/O (fgets(), fputs())
  - 9.5.3 Formatted I/O (fscanf(), fprintf())
  - 9.5.4 Record I/O (fwrite(), fread())
- 9.6 Overview of random file access
- 9.7 Error handling

## **10 Recent Trends in Programming**

**(2 hours)**

- 10.1 Introduction to object oriented programming (OOP)
- 10.2 Definitions of class, method and object in OOP
- 10.3 Difference between procedure oriented and OOP
- 10.4 Overview of other high level programming languages

## **Practical**

**(45 hours)**

- 1. Lab 1: Introduction and demonstrations of projects written in C
- 2. Lab 2: Formatted and unformatted Input/output in C
- 3. Lab 3: Branching in control structure
- 4. Lab 4: Looping in control structure
- 5. Lab 5: Array in C
- 6. Lab 6: String in C
- 7. Lab 7: Pointers in C
- 8. Lab 8: User defined functions in C
- 9. Lab 9: Structure in C
- 10. Lab 10: File handling in C
- 11. Group project on C maximum 4 students in a group at the end of the course.

## **Reference**

- 1. Lafore, R. (1993). *The Waite Group's C programming using Turbo C++* (1st ed.). SAMS Publishing.
- 2. Balagurusamy, E. (2019). *Programming in ANSI C* (8th ed.). McGraw Hill Education.
- 3. Gottfried, B. S. (2010). *Programming with C* (3rd ed.). McGraw Hill Education.