

## The “C” Programming Language

**Duration: 40 Hours**

**Course description:** The course fully covers the basics of programming in the “C” programming language and demonstrates fundamental programming techniques, customs and vocabulary including the most common library functions and the usage of the preprocessor.

**Learning objectives:** To familiarize the trainee with basic concepts of computer programming and developer tools. To present the syntax and semantics of the “C” language as well as data types offered by the language To allow the trainee to write their own programs using standard language infrastructure regardless of the hardware or software platform

### **Course outline**

- Introduction to compiling and software development
  - Basic scalar data types and their operators
  - Flow control
  - Complex data types: arrays, structures and pointers
  - Structuring the code: functions and modules
  - Preprocessing source code
- =====

### **Introduction**

The C Language and its Advantages  
The Structure of a C Program

### **Writing C Programs**

Building an Executable Version of a C Program  
Debugging a C Program  
Examining and Running a C Application Program

### **Data Types and Variables**

Data Types  
Operands, Operators, and Arithmetic Expressions

### **Input/Output Management**

The Input/Output Concept  
Formatted Input Function

### **Control-Flow Statements**

The Control-Flow Program Statements

Looping Statements

The Data-checking process

### **Modular Programming with Functions**

The C Function

Passing Data to Functions

Passing an Address to Modify a Value in Memory

Using Functions in the Checkbook Program

C Standard Library Functions

### **Arrays, Pointers, and Strings**

Arrays , Pointers , Strings

Using Arrays, Strings, and Pointers in the Checkbook Program

### **Structures**

Structures

Arrays of Structures

Passing Structures to Functions

Nesting Structures

### **File Input/Output**

Command-line Arguments

Combining Command-line

## “C++ programming language”( 40 Hours)

**Objective:** At the end of the course, we expect people to have a good understanding about the concept of object-oriented programming using C++, be able to write and read basic C++ code.

**Prerequisite:** No prior knowledge about C++ is required, but people are expected to have some basic knowledge about computers, some knowledge about one or two other programming languages such as Perl, PHP, Python or Java etc is preferred.

### Course Outlines:

- Introduction to C++. Creating a project, Writing, compiling and running a program.
- Variables and data types.
- Expressions. Constants. Operators conversions.
- Looping constructs: while, do...while, for loops.
- If...else statements. Switch/case construct.
- Functions.
- Passing arguments. Function prototyping. Default argument initializers. Inline functions.
- Arrays.
- Array initialisation. Multi-dimensional arrays. Character arrays. Working with character strings.
- Storage Classes.
- Global variables.
- Pointers.
- Pointer and arrays. Pointers to character strings. Arrays of pointers. Memory slicing. Pointers to functions.
- C++ classes.
- Data members and member functions. Creating objects. The new and delete operators. Friends to a class. Class initialisation
- Reference types.
- Reference type arguments.
- Function overloading.
- Operator overloading.
- Copy constructor.
- Assignment operator.
- Template classes.
- Static class members. File streams.

- Inheritance.
- Base classes and derived classes. Inherited member access. Base class initialisation. Protected members of a class.
- Virtual functions.
- Virtual destructors.
- Virtual base classes.
- Virtual base class member access.
- Constructor and destructor ordering.
- Exception handling.
- try...throw...catch block. Nested catch handlers.