**C | C++ Programming**

**Course Overview**

The **C | C++ Programming** course is designed to build a strong foundation in programming using two of the most fundamental and powerful languages in the software industry — **C and C++**. Starting from the basics of structured programming in C, the course transitions into object-oriented programming with C++, preparing learners for modern software development. These languages are the backbone of system software, embedded systems, game development, and high-performance applications. Whether you're a beginner or want to strengthen your programming concepts, this course offers a robust pathway to becoming a proficient programmer.

**Course Objectives**

By the end of this course, learners will be able to:

* Understand the fundamentals of programming and structured coding in **C**.
* Develop logic-building skills using conditional statements, loops, arrays, and functions.
* Gain deep understanding of memory management, pointers, and data structures in C.
* Transition smoothly into **C++**, understanding the principles of Object-Oriented Programming (OOP).
* Apply concepts such as **classes, inheritance, polymorphism, abstraction, and encapsulation**.
* Build small- to medium-scale projects using both C and C++.
* Prepare for further learning in advanced technologies like Data Structures, Algorithms, or Game Development.

**Syllabus Breakdown**

**Module 1: Introduction to C Programming**

* History and features of C
* Structure of a C program
* Compilation and execution process
* Data types, variables, and constants
* Operators and expressions
* Input and output functions

**Module 2: Control Flow and Functions**

* Conditional statements: if, if-else, switch
* Looping structures: for, while, do-while
* Introduction to functions
* Function arguments and return values
* Recursive functions
* Scope rules: global vs local variables

**Module 3: Arrays, Strings, and Pointers**

* One-dimensional and multi-dimensional arrays
* String handling and manipulation
* Introduction to pointers
* Pointer arithmetic
* Pointers and functions
* Dynamic memory allocation (malloc, calloc, free)

**Module 4: Structures and File Handling in C**

* Defining and using structures
* Array of structures
* Nested structures
* File operations: reading, writing, appending
* File pointers and formatted I/O

**Module 5: Introduction to C++ Programming**

* Overview of C++ vs C
* Input/Output using cin and cout
* Object-Oriented Programming (OOP) concepts
* Classes and objects
* Constructors and destructors
* Inline functions and friend functions

**Module 6: Advanced C++ Concepts**

* Inheritance: single, multiple, multilevel
* Polymorphism: function overloading and operator overloading
* Virtual functions and runtime polymorphism
* Abstract classes and interfaces
* Templates: function and class templates
* Exception handling in C++

**Module 7: Projects and Practical Applications**

* Mini-project in C: e.g., Student Record System, Calculator
* Mini-project in C++: e.g., Bank Management System, Library Management
* Debugging and code optimization practices

**Career Opportunities**

Mastering C and C++ opens up multiple entry points in the tech industry. Some potential roles and paths include:

* **Software Developer / Programmer**
* **Embedded Systems Engineer**
* **Game Developer**
* **System Programmer**
* **Data Structures & Algorithms Specialist**
* **Firmware Engineer**
* **Competitive Programming and Coding Interview Preparation**

Additionally, C and C++ are often prerequisites for learning advanced areas like **Operating Systems**, **Compiler Design**, **Game Engines**, and **High-Frequency Trading Systems**. Strong command over these languages is also vital for **placements and coding contests** in top IT companies.