



Syllabus: Certificate in C++ Core

- **Duration:** 3 Months
 - **Eligibility:** 10th Pass (Knowledge of C Programming recommended)
-

Evaluation Scheme

- **Full Marks:** 200
 - **Theory:** 100 Marks
 - **Practical/Project Works:** 80 Marks
 - **Internal Assessment/Viva (Oral Test):** 20 Marks
-

Course Syllabus

This course introduces the principles of Object-Oriented Programming (OOP) using C++, building upon the procedural concepts of C to develop more robust and scalable applications.

Module 1: Introduction to C++ and OOP

- Principles of Object-Oriented Programming: Encapsulation, Abstraction, Inheritance, Polymorphism.
- C++ vs. C: Key differences and advantages.
- Structure of a C++ Program.
- Basic Input/Output using cin and cout.
- Data Types, Keywords, and Operators.

Module 2: Classes, Objects, and Functions

- **Classes and Objects:** Defining classes, creating objects, and accessing members.
- Member Functions: Inside and outside class definitions.
- **Constructors and Destructors:** Default, Parameterized, and Copy constructors.
- Function Overloading.

Module 3: Inheritance and Polymorphism

- **Inheritance:** Base and Derived classes, types of inheritance (Single, Multiple, Hierarchical).
- Access Specifiers: public, private, protected.
- **Polymorphism:** Compile-time (Function Overloading) and Run-time (Virtual Functions).
- Operator Overloading.



Module 4: Advanced C++ Concepts

- **Templates:** Function and Class templates for generic programming.
- **Exception Handling:** Using try, catch, and throw blocks to manage errors.
- Introduction to the Standard Template Library (STL): Containers, Iterators, and Algorithms.

Module 5: File Handling & Final Project

- Working with files: Opening, reading, writing, and closing files.
- **Final Project:** Develop a console-based application using OOP principles.
- **Example Project:** A "Bank Account Management System" or "Employee Management System" that uses classes, objects, and file I/O to manage records.

