



# PROFESSIONAL C++

## Description

This course will give the student to object-oriented programming concepts of program specification and design, coding and testing using a modern software development environment. Students will also learn how to write programs in an object-oriented high level programming language. Topics covered include fundamentals of algorithms, flowcharts, problem solving, OOPS programming concepts, classes and methods, control structures, arrays, and strings. Throughout the session, problem solving skills will be stressed and applied to solving computing problems. Weekly laboratory experiments will provide hands-on experience in topics covered in this course.

## Expectations and Goals

The learning goals of this course are:

- To understand how C++ improves C with object-oriented features.
- To learn how to write inline functions for efficiency and performance.
- To learn the syntax and semantics of the C++ programming language.
- To learn how to design C++ classes for code reuse.
- To learn how to implement copy constructors and class member functions.
- To understand the concept of data abstraction and encapsulation.
- To learn how to overload functions and operators in C++.
- To learn how containment and inheritance promote code reuse in C++.
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
- To learn how to design and implement generic classes with C++ templates.
- To learn how to use exception handling in C++ programs.

## Prerequisites

Basic Computing and Logical Knowledge, Knowledge of C Programming.



Module	Topic
<b>Module 1</b>	<b>Introduction</b> Introduction to C++. Procedural vs. Object Oriented Programming(OOP) Benefits of OOPs Different OOPs Features Basic Components of C++ Compiling and Executing C++ program
<b>Module 2</b>	<b>Fundamental of C++</b> Tokens, Keywords, Identifiers and Constants Data Types, Type Compatibility and Variables Operators in C++ Operator precedence Control Statement. Iteration and Loops
<b>Module 3</b>	<b>Function in C++</b> Type of Function, Function Prototyping Call by Reference and Call by value Scope and Visibility of variables in Functions Inline Function, Friend Function
<b>Module 4</b>	<b>Variadic Function in C++</b> What is Variadic Function Use of Variadic Function Types of Variadic Function A C++ Program to implement a variadic function
<b>Module 5</b>	<b>Basic Concept in OOPs</b> Objects and Classes Encapsulation Abstraction This pointer Polymorphism Inheritance Dynamic Binding Message Passing
<b>Module 6</b>	<b>Object and Classes</b> Access Specifier Specifying a Class and Create an Object Defining Member Function A C++ program with Class
<b>Module 7</b>	<b>Constructors and Destructors</b> Default Constructor, Parameterized Constructor, Copy Constructor, Dynamic Constructor Constructor Overloading How to define a Destructor

<b>Module 9</b>	<b>Polymorphism</b> What is Polymorphism Run-time and Compile-time Polymorphism Function Overloading Operator Overloading Virtual Function
<b>Module 10</b>	<b>Files and Exception Handling</b> Classes for File Stream Operations Opening and Closing a File File Modes, File Pointers Input-Output Operations Updating a File Types of Error and Exceptions Try-Catch-Throw mechanism
<b>Module 11</b>	<b>Templates</b> Template Class Template Function Implementation of Templates using C++
<b>Module 12</b>	<b>Standard Template Library (STL)</b>
<b>Module 13</b>	Project work and documentation



Scan This QR code to  
Download the ARDENT-ian App



**Head Office :**

SDF Building, Module 132, Ground Floor, Salt Lake  
Sector V Kolkata: 700091

**Toll-Free: 1800-313-4533**