Document Generated: 12/18/2025

Learning Style: On Demand

Technology: CSS

Difficulty: Intermediate

Course Duration: 5 Hours

C++ Intermediate

About this Course:

What is C++

C++ is a general-purpose object-oriented programming (OOP) language, developed by Bjarne Stroustrup, and is an extension of the C language. It is therefore possible

to code C++ in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an effective example of a hybrid language.

C++ is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features. Initially, the language was called "C with classes" as it had all the properties of the C language with an additional concept of "classes." However, it was renamed C++ in 1983.

The C++ Intermediate course is the second course in the two-course series on C++. The course is an intermediate level course that builds on the basic C++ coding skills taught in the C++ Fundamentals course. The course introduces the concepts of Standard Template Library (STL), data structures, iterators, sequence containers and function objects. The C++ Intermediate is a programming course focused on the advanced concepts of C++ coding. The course has multiple hands-on programming exercises to enable the students to apply the learnt concepts while working with C++ programming language. The course is essential to develop intermediate programming skills in the students.

The main highlight of C++ is a collection of predefined classes, which are data types that can be instantiated multiple times. The language also facilitates declaration of user-defined classes. Classes can further accommodate member functions to implement specific functionality

Course Outline:

Chapter 01 - Data Structures and Pointers

1. Topic A: Classes and Structs - Part 1
2. Classes and Structs - Part 2
3. Classes and Structs - Part 3
4. Topic B: Enums - Part 1
5. Enums - Part 2
6. Enums - Part 3
7. Topic C: Unions - Part 1
8. Unions - Part 2
9. Unions - Part 3
10. Topic D: Typedefs - Part 1
11. Typedefs - Part 2
12. Typedefs - Part 3
13. Topic E: Introduction to Pointers - Part 1
14. Introduction to Pointers - Part 2
15. Introduction to Pointers - Part 3
16. Topic F: Pointers and Array Indexing - Part 1
17. Pointers and Array Indexing - Part 2
18. Pointers and Array Indexing - Part 3
19. Topic G: Using Const with Pointers - Part 1
20. Using Const with Pointers - Part 2
21. Using Const with Pointers - Part 3
22. Topic H: Pointers to String Literals - Part 1

23. Pointers to String Literals - Part 2
24. Pointers to String Literals - Part 3
25. Topic I: References - Part 1
26. References - Part 2
27. References - Part 3
28. Topic J: Smart Pointers - Part 1
29. Smart Pointers - Part 2
30. Smart Pointers - Part 3

Chapter 02 - Arrays and Strings

1. Topic A: Arrays - Part 1
2. Arrays - Part 2
3. Arrays - Part 3 Topic B: Standard Library Strings - Part 1
4. Standard Library Strings - Part 2
5. Standard Library Strings - Part 3
6. Topic C: More Standard Library Strings - Part 1 More Standard Library Strings - Part 2
7. More Standard Library Strings - Part 3
8. Topic D: Functions - Part 1
9. Functions - Part 2
10. Functions - Part 3
11. Topic E: More Functions - Part 1
12. More Functions - Part 2
13. More Functions - Part 3 Topic F: Function Pointers - Part 1
14. Function Pointers - Part 2
15. Function Pointers - Part 3
16. Topic G: Control Statements - Part 1

17. Control Statements - Part 2 Control Statements - Part 3