



Document Generated: 12/22/2024

Learning Style: Virtual Classroom

Provider:

Difficulty: Intermediate

Course Duration: 4 Days

Next Course Date: April 14, 2025

Intermediate C++ 20 Programming

About This Course:

C++ is a powerful, high-performance programming language that offers an ideal blend of low-level memory manipulation and high-level abstraction capabilities. Learning C++ is a valuable investment for developers, as it opens the door to creating efficient, versatile, and complex applications that run on a variety of platforms. Modern companies across diverse industries – including finance, gaming, automotive, and telecommunications – rely on C++ for developing performance-critical applications, system software, and embedded systems. Renowned organizations like Google, Facebook, and Microsoft continue to leverage the power of C++ in their development practices, solidifying its status as a crucial skill for developers seeking lucrative and challenging career opportunities.

Course Objectives:

- Master intermediate to advanced C++ 20 programming techniques, enabling the development of efficient and maintainable applications using the latest features and best practices.
- Acquire in-depth knowledge of memory management in C++, including the handle/body pattern, smart pointers, and move constructors, to optimize performance and minimize memory-related issues.
- Develop proficiency in functional programming with C++, incorporating concepts such as dependency injection, functors, and lambda expressions to enhance code flexibility and modularity.
- Gain expertise in utilizing the C++ Standard Library for generic programming, mastering the use of containers, algorithms, numerics, and other features to create powerful, reusable code components.
- Learn to implement effective unit testing in C++ using GTest, ensuring the reliability and robustness of your applications through rigorous testing methodologies.
- Understand the basics of multitasking in C++, exploring threads, tasks, and async for concurrent programming, empowering developers to create scalable and high-performance applications.

Audience:

- This is an intermediate level development course designed for developers with prior C++ programming experience

Prerequisites:

- Students without prior C++ programming background should take the pre-requisite training.

Take Before: Incoming students should have practical skills equivalent to the topics in, or should have recently attended, one of these courses as a pre-requisite:

- TTCP2100: [Introduction to C++ Programming](#)

Course Outline:

- Master intermediate to advanced C++ 20 programming techniques
- Acquire in-depth knowledge of memory management in C++,
- Develop proficiency in functional programming with C++

- Gain expertise in utilizing the C++ Standard Library for generic programming
- Learn to implement effective unit testing in C++ using GTest
- Understand the basics of multitasking in C++, exploring threads, tasks, and async for concurrent programming