

INTERMEDIATE C++ 20 PROGRAMMING | EFFECTIVE C++ 20 (TTCP2150)

Course Code: 7390

Explore C++ Templates, Memory Management, Functional Programming, Unit Testing & Modern Features & More

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.

Geared for experienced C++ developers, Intermediate C++ 20 / Effective C++ 20 is a 4 day, hands-on program that dives covers a broad spectrum of topics - from the quick review of C++ essentials to modern C++ features, memory management, unit testing, and more. Our expert instructors will walk you through a comprehensive journey, investigating cutting-edge concepts such as RAII, copy and move semantics, namespaces, templates, and C++ 20 Concepts & auto Templates. You'll learn to leverage the power of modern C++ and unravel the intricacies of memory management, including the handle/body pattern, smart pointers, and move constructors. By the end of this course, you'll have an in-depth understanding of C++ memory, pointers, and complexity.

Working in a hands-on environment, explore the art of functional programming and discover how the IoC pattern, dependency injection, functors, and lambda expressions can bring about significant enhancements to your code. With a strong emphasis on SOLID principles, inheritance, polymorphism, exceptions, and operator overloading, this course will help you design robust, maintainable, and scalable modern applications. You'll also expand your C++ toolset by exploring the rich offerings of the Standard Library, mastering the essentials of containers, algorithms, numerics, dates, and times. Gain a solid introduction to multitasking with threads, tasks, and async. As a bonus, you'll also learn how to implement effective unit testing in C++ using GTest, ensuring your code is reliable and bug-free.

Join our immersive training experience and become an adept C++ developer with

unparalleled skills in the latest C++ 20 programming techniques. This fast-paced, lab-intensive course is designed to equip you with the knowledge and confidence to tackle the most challenging C++ development projects.

What You'll Learn

Working in a hands-on learning environment, guided by our expert team you'll learn to:

- 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.

Who Needs to Attend

This is an **intermediate level** development course designed for developers with prior C++ programming experience. Students without prior C++ programming background are also eligible to enroll in this course provided they take the pre-requisite training.

Prerequisites

This is an intermediate level development course designed for developers with prior C++ programming experience. Students without prior C++ programming background should take the pre-requisite training.

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

INTERMEDIATE C++ 20 PROGRAMMING | EFFECTIVE C++ 20 (TTCP2150)

Course Code: 7390

CLASSROOM LIVE

\$2,595 USD

4 Day

Classroom Live Outline

1. **Quick Review of C++**

- Implementing a basic O-O design
- Implementing Classes
- Visibility & friends
- File organization
- C++ types - structs, classes, interfaces, enums

2. **Modern C++**

- New features in C++ 11,14,17,20
- RAII - Modern memory management in C++ - overview
- Copy vs Move semantics
- Namespaces
- Strings
- Input & Output
- Implementing a linked-list - a demonstration of class, memory, pointers and complexity

3. **Templates**

- General Purpose Functions
- Function Templates
- Template Parameters
- Template Parameter Conversion
- Function Template Problem
- Generic Programming
- General Purpose Classes
- Class Templates
- Class Template Instantiation
- Non-Type Parameter
- C++ Containers overview
- C++ 20 concepts & auto Templates

4. **Memory Management**

- The handle/body (Bridge) pattern
 - Using strings effectively
 - Smart Pointers
 - Move constructor in depth
 - Other <memory> features
5. **Unit Testing in C++**
- Unit testing - Quick Overview
 - Unit testing in C++
 - Using GTest
6. **Inheritance and Polymorphism**
- Inheritance Concept
 - Inheritance in C++
 - Virtual Function Specification
 - Invoking Virtual Functions
 - VTable
 - Virtual Destructors
 - Abstract Class Using Pure Virtual Function
 - Design for Polymorphism
 - Interfaces
 - Design for Interface
 - A SOLID introduction
7. **Exceptions**
- Review of the basics: try, catch, throw
 - The throws declaration in modern C++
 - Using noexcept
 - Overriding terminate
8. **Operator Overloading & Conversion**
- Basics
 - Essential Operators
 - Conversion Operators
 - Constructor as conversion
 - Explicit vs Implicit conversion
9. **Functional Programming**
- The IoC pattern
 - Dependency Injection
 - Functions as objects
 - IoC via interface
 - Functors
 - IoC with Functors
 - Implementing Functors
 - Function Pointers
 - IoC with Function Pointers
 - Lambda Expressions
 - Lambda Syntax
 - IoC with Lambdas

10. **Standard Library**

- Perspective
- History and Evolution
- New Features
- Generic Programming
- Containers
- Algorithms
- Numerics
- Dates & Times
- Initializer List

11. **Introduction to Multitasking**

- Threads
- Tasks
- Async

INTERMEDIATE C++ 20 PROGRAMMING | EFFECTIVE C++ 20 (TTCP2150)

Course Code: 7390

VIRTUAL CLASSROOM LIVE

\$2,595 USD

4 Day

Virtual Classroom Live Outline

1. **Quick Review of C++**

- Implementing a basic O-O design
- Implementing Classes
- Visibility & friends
- File organization
- C++ types - structs, classes, interfaces, enums

2. **Modern C++**

- New features in C++ 11,14,17,20
- RAII - Modern memory management in C++ - overview
- Copy vs Move semantics
- Namespaces
- Strings
- Input & Output
- Implementing a linked-list - a demonstration of class, memory, pointers and complexity

3. **Templates**

- General Purpose Functions
- Function Templates
- Template Parameters
- Template Parameter Conversion
- Function Template Problem
- Generic Programming
- General Purpose Classes
- Class Templates
- Class Template Instantiation
- Non-Type Parameter
- C++ Containers overview
- C++ 20 concepts & auto Templates

4. **Memory Management**

- The handle/body (Bridge) pattern
 - Using strings effectively
 - Smart Pointers
 - Move constructor in depth
 - Other <memory> features
5. **Unit Testing in C++**
- Unit testing - Quick Overview
 - Unit testing in C++
 - Using GTest
6. **Inheritance and Polymorphism**
- Inheritance Concept
 - Inheritance in C++
 - Virtual Function Specification
 - Invoking Virtual Functions
 - VTable
 - Virtual Destructors
 - Abstract Class Using Pure Virtual Function
 - Design for Polymorphism
 - Interfaces
 - Design for Interface
 - A SOLID introduction
7. **Exceptions**
- Review of the basics: try, catch, throw
 - The throws declaration in modern C++
 - Using noexcept
 - Overriding terminate
8. **Operator Overloading & Conversion**
- Basics
 - Essential Operators
 - Conversion Operators
 - Constructor as conversion
 - Explicit vs Implicit conversion
9. **Functional Programming**
- The IoC pattern
 - Dependency Injection
 - Functions as objects
 - IoC via interface
 - Functors
 - IoC with Functors
 - Implementing Functors
 - Function Pointers
 - IoC with Function Pointers
 - Lambda Expressions
 - Lambda Syntax
 - IoC with Lambdas

10. **Standard Library**

- Perspective
- History and Evolution
- New Features
- Generic Programming
- Containers
- Algorithms
- Numerics
- Dates & Times
- _INITIALIZER List

11. **Introduction to Multitasking**

- Threads
- Tasks
- Async

Virtual Classroom Live Labs

This course is about 50% hands-on lab and 50% lecture, with extensive programming exercises designed to reinforce fundamental skills and concepts taught throughout the course. Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review. This course is “skills-centric”, designed to train attendees in essential object oriented, intermediate and beyond level C++ development skills, coupling the most current, effective techniques with the soundest coding practices.



INTERMEDIATE C++ 20 PROGRAMMING | EFFECTIVE C++ 20 (TTCP2150)

Course Code: 7390

PRIVATE GROUP TRAINING

4 Day

Visit us at www.globalknowledge.com or call us at 1-866-716-6688.

Date created: 4/23/2026 9:04:36 AM

Copyright © 2026 Global Knowledge Training LLC. All Rights Reserved.