

Intermediate C++ 20 Programming | Effective C++ 20 (TTCP2150)

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

Course Snapshot

- **Course:** Intermediate C++ 20 Programming | Effective C++20 (TTCP2150)
- **Duration:** 4 days
- **Audience & Skill Level:** This is an intermediate and beyond level course geared for experienced C++ programmers who have current, hands-on C++ coding experience. This course is not for non-developers, or for developers new to C++.
- **Hands-on:** This course combines engaging instructor-led presentations and practical demonstrations with hands-on programming exercises, challenge labs, use case exploration and engaging group activities. Student machines are required.
- **Format:** This course can be delivered for your team or organization **online live (virtual)**, **onsite in-person** or across our robust **blended learning experience platform (LXP)**.
- **Public Schedule:** This course is currently available on our Public Open Enrollment Schedule.
- **Customizable:** This course agenda, topics, labs, hours and delivery modalities can be adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.
-

Overview

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 four 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.

Learning Objectives

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.

Audience & Pre-Requisites

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.

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

Related Training | C++ Training Series

The following is a small subset of our related courses. Please see our full catalog for a complete list.

- TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers (TTCODE101-C / TTCP2000)
- TTCP2100 Introduction to C++ Programming | C++ Essentials
- TTCP2150 Intermediate C++ Programming | Effective C++ Programming
- TTCP3103 Advanced C++ Programming
- TTCP1250 Object Oriented Design Patterns and Best Practices in C++
- TTCP1270 SOLID Design in C++

Follow On Courses: Our Skills Academy Developer Tracks include a wide variety of follow-on courses and learning paths for leveraging C++ for next-level development, testing, security and more. Please see our **C++ Developer Training Suite & Learning Paths** list of courses or inquire for recommendations based on your specific role and goals.

Course Setup / Tools Options: There are many options for the student development platform. IDE's from Visual Studio, Visual Studio Code, Eclipse and others are acceptable. A compiler compatible with C++ 20 is also required. GCC version 12+ or Visual Studio 2022 (MSVC) are compatible. GTest is also required for the course and comes with VS 2022. For Linux and other platforms, it must be installed. Students may configure their own environment or, ask us about a virtual lab setup with everything already installed

Enhanced Learning Services: Please also ask about our robust Learning Experience Platform (LXP), Skills Assessment & Skills Prep Services, Skills Immersion Programs & Camps, Coaching and Mentoring Services and Extended Learning Support programs.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill level, interests and participation.

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
- RAll - 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

Student Materials & Lab Environment

All course software (limited versions, for course use only), digital courseware files or course notes, labs / data sets and solutions (as applicable) are provided for you in our “easy access / no install required” high-speed remote lab environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience. Please ask for details.

Setup Made Simple with our Robust Learning Experience Platform (LXP)

All course software (limited versions, for course use only), knowledge checks, digital courseware files or course notes, labs / data sets and solutions, live coaching support channels (as applicable) and rich extended learning / post training resources are provided for you in our “easy access / no install required” high-speed **Learning Experience Platform (LXP)** remote lab and content environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience.

For More Information

For more information about our dedicated skills-focused training services (instructor-led, self-paced or blended), collaborative coaching services, robust Learning Experience Platform (LXP) solutions, Career Experiences, public course schedule, partner programs, courseware licensing options or to see our complete list of course offerings, training solutions and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**. Our pricing and services are always satisfaction guaranteed.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS • PUBLIC / OPEN ENROLLMENT COURSES
LEARNING EXPERIENCE PLATFORM (LXP) • COACHING / MENTORING • ASSESSMENTS • CONTENT LICENSING & DEVELOPMENT
LEARNING PLAN DEVELOPMENT • SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS
PARTNER & RESELLER PROGRAMS • CORPORATE TRAINING MANAGEMENT • VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm

