

Advanced C++ 20 Programming (TTCP2175)

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

Course Snapshot

- **Course:** Advanced C++ 20 Programming (TTCP2175)
- **Duration:** 3 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

Geared for experienced C++ developers, **Advanced C++ 20 Programming** is a three-day hands-on course designed to provide you with skills required to write faster, robust C++ code, enhancing your ability to create performance-critical applications ranging from system software, game development, to real-time systems and AI programming.

Working in a lab-focused learning environment guided by our experienced Instructor, you'll explore a broad spectrum of 'next-level' topics such as SOLID design principles, operator overloading, functional programming, and template usage, all aimed at refining your programming craft. The hands-on work will mirror real-world scenarios, including implementing design patterns and managing threads and tasks. You'll also discover the realms of multi-threading and asynchronous programming, invaluable skills for creating efficient, high-performance applications. These advanced skills have significant application in industries such as finance for high-frequency trading systems, in gaming for building high-performance game engines, or in tech companies for building large scale distributed systems.

By the end of this unique and intensive course, you will be well-equipped to tackle complex coding challenges, contribute more effectively to your team's projects, and deliver high-quality, efficient applications that meet modern business demands.

Learning Objectives

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

- **Become a Pro at SOLID Design:** You'll delve into SOLID design principles, mastering how to write clean, maintainable code. By the end, you'll be able to identify and avoid design smells, enhancing the overall quality of your projects.
- **Master the Art of Factory Implementation:** Get hands-on with factories in C++. We'll guide you through the basics and options, including Singleton, to help you understand the critical role of factories in object-oriented design.
- **Up Your Game with Operator Overloading:** You'll learn about operator overloading and its applications. By understanding how to enhance the readability and flexibility of your code, you'll streamline your programming tasks.
- **Unlock the Power of Templates:** We'll dive into the intriguing world of templates, exploring variance, concepts, and the 'auto' keyword. You'll get to implement covariant and contravariant templates, broadening your C++ expertise.
- **Ace Multithreading and Asynchronous Programming:** We'll explore the realms of multithreading and asynchronous programming, equipping you with the tools to create efficient, high-performance applications. You'll get to practice with mutexes, semaphores, atomics, and coroutines, gaining invaluable experience for your future projects.

Audience & Pre-Requisites

This is an **intermediate and beyond 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 prerequisite:

- TTCP2100 Introduction to C++ Programming / C++ 20

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++ 20 Programming | C++ Essentials
- TTCP2150 Intermediate C++ 20 Programming | Effective ++ Programming
- TTCP2175 Advanced C++ 20 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.

- | | | |
|--|---|--|
| <p>1. C++ Quick Review</p> <ul style="list-style-type: none"> • Modern C++ - an introduction to new features in C++ 11-20 | <ul style="list-style-type: none"> • Options • Singleton • A C++ Object Factory | <ul style="list-style-type: none"> • Implementing contravariant templates • <concepts> • auto |
| <p>2. SOLID Design</p> <ul style="list-style-type: none"> • Design Smells • Project Overview • Single Responsibility • Open/Close • Liskov's Substitution • Interface Segregation • Dependency Inversion | <p>4. Operator Overloading</p> <ul style="list-style-type: none"> • Commonly Overloaded operators • Conversions • Constructor Conversions • Implicit vs Explicit | <p>6. Functional Programming</p> <ul style="list-style-type: none"> • Lambda Expressions • Functors • <functional> |
| <p>3. Implementing a Factory in C++</p> <ul style="list-style-type: none"> • Factory Basics | <p>5. Templates</p> <ul style="list-style-type: none"> • Understanding variance • Implementing covariant templates | <p>7. Structural Patterns</p> <ul style="list-style-type: none"> • Adapter • Bridge • Composite • Decorator |

- RAII and Proxy Pattern – Smart Pointers
 - Strategies for Smart Pointers & Raw Pointers
 - Other patterns
- 8. Behavioral Patterns**
- Solving common design smells with behavioral patterns
- Template Method – issues initializing C++ objects
 - State Pattern
 - Strategy Pattern
 - Command Pattern
 - Other behavioral Patterns
- 9. Threads, Tasks, Async**
- All about threads
- Mutex
 - Semaphores
 - Latch & barrier
 - atomics
 - All about Tasks
 - <future>
 - Coroutines (async)

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

All applicable course software, digital courseware files or course notes, labs, data sets and solutions, live coaching support channels and rich extended learning and 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. Course materials, software, resources and post-training platform access periods vary by course.

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

