Advanced C++ 20 Programming (Modern C++)

Duration: *3 Day(s)*

Course Overview

Geared for experienced C++ developers, Advanced C++ 20 Programming / Modern C++ 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.

Review this course online at https://www.alta3.com/courses/TTCP2175

Objectives

- Master SOLID design principles for clean, maintainable code.
- Implement complex C++ design patterns and factories.
- Enhance code with operator overloading and templates.
- Develop efficient, high-performance applications with multithreading and asynchronous programming.

Who Should Attend

- Experienced C++ Developers
- System Software Engineers
- Game Development Specialists
- AI Programmers

Prerequisites

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

Course Outline

C++ Quick Review

1. Modern C++ - an introduction to new features in C++ 11-20

SOLID Design

2. Design Smells

3. Project Overview
4. Single Responsibility
5. Open/Close
6. Liskov’s Substitution
7. Interface Segregation
8. Dependency Inversion

Implementing a Factory in C++

9. Factory Basics
10. Options
11. Singleton
12. A C++ Object Factory

Operator Overloading

13. Commonly Overloaded operators
14. Conversions
15. Constructor Conversions
16. Implicit vs Explicit

Templates

17. Understanding variance
18. Implementing covariant templates
19. Implementing contravariant templates
20.
21. auto

Functional Programming

22. Lambda Expressions
23. Functors
24.

Structural Patterns

25. Adapter
26. Bridge
27. Composite
28. Decorator
29. RAII and Proxy Pattern – Smart Pointers
30. Strategies for Smart Pointers & Raw Pointers
31. Other patterns

Behavioral Patterns

32. Solving common design smells with behavioral patterns
33. Template Method – issues initializing C++ objects
34. State Pattern
35. Strategy Pattern
36. Command Pattern
37. Other behavioral Patterns

Threads, Tasks, Async

38. All about threads
39. Mutex
40. Semaphores
41. Latch & barrier
42. atomics
43. All about Tasks
44.
45. Coroutines (async)