Advanced C++ 20 Programming / Modern C++ (TTCP2175) Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Overview

Advanced C++ 20 Programming (Modern C++) is a hands-on training designed to help developers write faster, more robust, and high-performance C++ applications. The course covers a wide range of applications including system software, game development, real-time systems, and AI programming.

Through instructor-led labs and real-world scenarios, participants will gain deep expertise in modern C++ programming techniques, design patterns, and performance optimization.

Who Should Attend

- Intermediate to advanced C++ developers
- Software engineers working on performance-critical systems
- Developers in gaming, AI, and system programming

What You Will Learn

By the end of this course, you will be able to:

- Apply SOLID design principles effectively
- Implement Factory and Singleton patterns
- Use operator overloading for flexible coding
- Master templates and modern C++ features
- Work with multithreading and async programming
- Use mutexes, semaphores, atomics, and coroutines
- Build real-world high-performance applications

Outline

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. For private delivery classes, 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. **C++ Quick Review**
   - Modern C++ - an introduction to new features in C++ 11-20
2. **SOLID Design**
   - Design Smells
   - Project Overview
   - Single Responsibility
   - Open/Close

BİLGİNÇ IT Academy

info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

- Liskov's Substitution
- Interface Segregation
- Dependency Inversion

3. **Implementing a Factory in C++**
   - Factory Basics
   - Options
   - Singleton
   - A C++ Object Factory
4. **Operator Overloading**
   - Commonly Overloaded operators
   - Conversions
   - Constructor Conversions
   - Implicit vs Explicit
5. **Templates**
   - Understanding variance
   - Implementing covariant templates
   - Implementing contravariant templates
   - <concepts>
   - auto
6. **Functional Programming**
   - Lambda Expressions
   - Functors
   - <functional>
7. **Structural Patterns**
   - 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)