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

Süre: **3 Gün**

BİLGİNÇ
IT Academy

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

Overview

Advanced C++ 20 Programming (Modern C++) eğitimi, geliştiricilerin daha hızlı, güvenilir ve yüksek performanslı C++ uygulamaları geliştirmesi için tasarlanmış ileri seviye, uygulamalı bir eğitimdir. Sistem yazılımlarından oyun geliştirmeye, gerçek zamanlı sistemlerden yapay zekâ uygulamalarına kadar geniş bir kullanım alanını kapsar.

Bu eğitimde, deneyimli eğitmenler eşliğinde gerçek dünya senaryolarına dayalı laboratuvar çalışmalarıyla; modern C++ teknikleri, tasarım desenleri ve performans optimizasyonu konularında uzmanlık kazanırsınız.

Who Should Attend

- Orta ve ileri seviyede C++ geliştiriciler
- Performans odaklı uygulama geliştiren yazılımcılar
- Sistem, oyun ve AI geliştiricileri

Not: Temel C++ bilgisi olmayan katılımcılar için ön eğitim önerilir.

What You Will Learn

Eğitim sonunda aşağıdaki konularda ileri düzey yetkinlik kazanacaksınız:

- SOLID prensipleri ile temiz ve sürdürülebilir kod yazımı
- Factory ve Singleton gibi tasarım desenleri
- Operator overloading ile esnek kod yapıları
- Template programlama ve modern C++ özellikleri
- Multithreading ve asynchronous programming
- Mutex, semaphore, atomic ve coroutine kullanımı
- Gerçek dünya senaryolarına uygun uygulama geliştirme

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

BİLGİNÇ IT Academy

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

    - Project Overview
    - Single Responsibility
    - Open/Close
    - 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)