



ADVANCED C++ PROGRAMLAMA EĞİTİMİ 5 GÜN



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz?	3
Ön Koşullar	4
Kimler Katılmalı.....	4
Outline	5
Modül 1	5
Modül 2	5
Modül 3	6
Modül 4	6
Modül 5	6

Eğitim Hakkında

Advanced C++ Programlama Eğitimi, C++'ın ileri düzey özelliklerini kapsamlı bir şekilde sunar. Katılımcılar, nesne yönelimli programlama (OOP) prensipleri üzerinde çalışırlar. Fonksiyonel programlama, modern C++ yapıları ve teknikleri ele alınır. STL (Standard Template Library), güçlü veri yapıları ve algoritmalarıyla tanışır. Tasarım şablonları, yazılım tasarımını iyileştirmek için kullanılır. Eşzamanlılık (concurrency), performansı artırmak için çoklu iş parçacığı yönetimini kapsar. Performans iyileştirme teknikleri, hız ve verimlilik açısından önemlidir.

Optimizasyon teknikleri, C++ kodunun performansını artırır. Bellek yönetimi, hafıza kullanımını optimize ederek, performansı artırır. C++ standart kütüphaneleri, tekrar kullanılabilir kod parçacıklarını içerir. İlgili teknolojiler, C++ programlama becerilerini genişletir. Pratik deneyim, gerçek dünya uygulamalarıyla becerileri pekiştirir.

Modern C++ özellikleri, kod kalitesini ve bakımını iyileştirir. Smart pointer'lar, otomatik bellek yönetimi sağlar. Lambda fonksiyonları, kodun daha sade olmasına yardımcı olur. Rvalue referansları, performansı artıran hareket semantiği sağlar. Algoritma ve veri yapıları, problemleri etkili bir şekilde çözmeye yardımcı olur. Template metaprogramming, derleme zamanında kod optimizasyonu sağlar. Exception handling, hataları güvenli bir şekilde ele alır.

Eşzamanlı programlama, karmaşık işlemleri paralel hale getirir. Thread'ler, işlemcileri daha etkili kullanır. Mutex ve lock'lar, veri yarışlarını önler. Atomic operations, thread-safe operasyonlar sağlar. Design patterns, yaygın problemler için çözümler sunar. Singleton, Factory ve Observer, popüler design pattern'lerdir.

C++'da veri yapıları, veriyi organize etme ve işleme yöntemleridir. Vector, list ve map yapıları sıklıkla kullanılır. Fonksiyonlar, kodu modüler hale getirir ve yeniden kullanılabilirlik sağlar. Polymorphism ve inheritance, OOP'nin temel taşlarıdır.

Son olarak, performans analizi ve optimizasyon stratejileri ele alınır. Profiling araçları, performans darboğazlarını tespit eder. Inline fonksiyonlar ve loop optimizasyonları, performansı artırır. Etkili bellek yönetimi, hız ve stabiliteyi iyileştirir.

Kısacası, Advanced C++ Programlama Eğitimi, C++ programlama dilinin ileri düzey yönlerini anlama ve uygulama becerisi kazandırır. Programcılar, bu eğitimle, modern ve etkili C++ uygulamaları geliştirme konusunda derinlemesine bilgi sahibi olurlar.

Neler Öğreneceksiniz?

Advanced C++ Programming eğitimi sırasında, aşağıdaki konuların bir kısmını öğrenebilirsiniz:

- Nesne Yönelimli Programlama (OOP): Sınıflar, nesnelere, inheritance (kalıtım), polymorphism (çok biçimlilik) ve diğer OOP kavramlarını öğreneceksiniz.
- Temel Veri Yapıları: Linked Lists, Stacks, Queues, Trees ve diğer veri yapılarını öğreneceksiniz.



- Bellek Yönetimi: Dynamic Memory Allocation, Stack and Heap bellek, smart pointers ve diğer bellek yönetim tekniklerini öğreneceksiniz.
- Fonksiyonlar ve Algoritmalar: Fonksiyonların yapısı, recursion (özyineleme), sorting (sıralama) ve diğer algoritmaları öğreneceksiniz.
- Standard Template Library (STL): Vector, List, Map, Set ve diğer STL sınıflarını kullanarak programlamayı optimize etmeyi öğreneceksiniz.
- İleri Seviye Nesne Yönelimli Programlama: Design Patterns, Exception Handling, Namespaces ve diğer ileri seviye OOP konularını öğreneceksiniz.

Bu liste tamamı değildir ve eğitim verilen konular eğitmen ve eğitim programına göre değişebilir.

Ön Koşullar

Advanced C++ Programming eğitimine katılmak için genellikle aşağıdaki ön koşulların yerine getirilmiş olması beklenir:

- Temel programlama deneyimi: Katılımcılar, temel programlama kavramlarını ve diğer programlama dillerinde programlama yapma deneyimine sahip olmalıdır.
- C++ deneyimi: Katılımcılar, C++ programlama dilinin temel kavramlarını ve kullanımını öğrenmiş ve pratik yapmış olmalıdır.
- Algoritma ve veri yapıları deneyimi: Katılımcılar, veri yapılarının ve algoritmaların temel kavramlarını ve kullanımını öğrenmiş ve pratik yapmış olmalıdır.

Bu ön koşullar, eğitim verilen konulara ve eğitmenin beklentilerine göre değişebilir ve bu nedenle eğitim verilen yerden doğrudan bilgi alınması önerilir.

Kimler Katılnmalı

Advanced C++ Programming eğitimine katılmak isteyen kişiler şunlar olabilir:

- Programlama deneyimi olan kişiler: Temel programlama deneyimi olan ve C++ programlama dilinde ileri seviyede programlamaya ilgi duyan kişiler.
- Gelişmiş C++ programcıları: C++ programlama dilini iyi bir şekilde kullanabilen ve bu dilin daha ileri seviyede özelliklerini ve kavramlarını öğrenmek isteyen programcıları.
- Veri yapıları ve algoritmalar hakkında bilgi sahibi olan kişiler: Veri yapıları ve algoritmalar hakkında temel bilgi sahibi olan ve bu bilgiyi C++ dilinde kullanma becerisini geliştirmek isteyen kişiler.
- İşletmeler ve organizasyonlar: İşletmeler ve organizasyonlar, çalışanlarının C++ programlama becerilerini geliştirmek ve ileri seviyede uygulamalar yapabilmelerini sağlamak için bu eğitimi tercih edebilirler.

Bu liste tamamı değildir ve herkes Advanced C++ Programming eğitimine katılma potansiyeline sahip olabilir, ancak bu eğitimin size uygun olup olmayacağına karar vermede eğitim verilen konulara, eğitim verilen yere ve katılımcının mevcut programlama deneyimine dikkat etmeniz önerilir.



Outline

Modül 1

- Intro + References
- Basics
- Workshop Basics
- Overloading
- What can we overload, and how.
- Extra C++ Types (bool & reference)
- Workshop overloading
- OOP
- Quick introduction to Oop
- Classes
- Structs
- Access Modifiers
- Constructor
- default/delete functions
- initializer syntax / constructor initialization list
- Workshop classes
- Memory
- Classical memory interaction
- Workshop Memory

Modül 2

- Inheritance
- Construction
- Polymorphism
- Virtual, pure virtual, abstract, interface
- Access modifiers
- Workshop Inheritance (Shapes)
- Exceptions
- What are they
- How do they work
- What to throw and what to catch
- Workshop exceptions
- Memory exhaustion
- How it's notified
- How to handle
- Modern Memory Management
- RAII
- Templates applied to Modern Memory Management (SmartPointer)



- Standardized C++11 SmartPointers
- Nullptr
- Workshop SmartPointers
- Namespaces
- Workshop Namespaces

Modül 3

- Auto
- The new auto keyword
- The new auto return syntax
- enum
- New style enums
- constexpr
- New constant expressions
- About constexpr
- Const and Mutable explained
- Lambdas & function objects
- Classes that act like functions
- Introduction lambda functions
- Chrono
- An introduction to the new Chrono library

Modül 4

- Casting
- Standard library
- String
- Containers
- Vector (vs)
- List
- Map
- Array
- Tuple
- Initializer lists
- Iterators
- range-for syntax
- Std Algorithms
- Streams
- Miscellaneous Keywords
- static
- explicit

Modül 5

- Move semantics



- Introduction to L/R values
- R-value-references applied to move semantics
- Type Traits
- Obtaining information on compile time
- Concurrency
- Introduction to C++11's Threading, async/future and atomic types implementation
- Variadic templates – An introduction to C++11's variadic templates