

C++ İin Giriş ve Orta Seviyede Dersler ve Kapsadıkları Konular

C++, programlama dillerinin en popüler ve güçlü örneklerinden biridir. 1979 yılında Bjarne Stroustrup tarafından geliştirilmiş, C dilinin genişletilmiş bir versiyonudur. C++'ın güçlü bir dil olmasının nedenleri arasında üst düzey performansı, sistem düzeyi programlama yetenekleri, nesne yönelimli yapısı ve kullanım alanlarının çeşitliliği bulunmaktadır.

Sololearn C++ giriş dersi, C++ programlama diline yeni başlayanlar için kapsamlı bir eğitim sağlıyor. Hemen başlamak için şuna tıklayın: [Giriş Dersi](#)

Ders içeriği şu şekilde özetlenebilir:

1. C++ Temelleri:

- Değişkenler, veri tipleri
- Operatörler ve ifadeler
- Denetim yapıları (if, switch, while, for adlı döngüler)

2. Fonksiyonlar:

- Fonksiyon tanımlama: fonksiyon türü, davranışı ve çağırılması
- Parametre geçişi
- Değer ve referans geçişi

3. Nesne Yönelimli Programlama:

- Sınıflar ve nesneler
- Erişim belirleyiciler
- Kurucu ve yıkıcı yöntemler

4. Standart Kütüphane ve Örnekler:

- Giriş/Çıkış İşlemleri
- Dizi (Array) ve Vektör Kullanımı
- Temel Algoritma Uygulamaları

Ders boyunca çok sayıda örnek kod ve pratik alıştırmaya içeriği bulunuyor. Böylece öğrenciler, C++'ın temel kavramlarını pratik yaparak pekiştirebiliyor.

Sololearn C++ giriş dersi, C++ programlamaya yeni başlayanlar için mükemmel bir başlangıç noktası. Derse katılanlar, C++'ın temel yapı taşlarını kavrayarak, sonraki daha gelişmiş konulara hazır hale geliyorlar.

Detaylı ders içeriği

Veri Tipleri ve Değişkenler:

C++'da temel veri tipleri olarak integer, float, double, char, bool gibi türler bulunur. Değişkenler tanımlanırken bu türler kullanılır. Ayrıca, diziler (array), string gibi daha karmaşık türler de mevcuttur.

Operatörler:

Aritmetik operatörler (+, -, *, /), atama operatörleri (=, +=, -=), karşılaştırma operatörleri (>, <, ==, !=) ve mantıksal operatörler (&&, ||, !) C++'da yaygın olarak kullanılır.

Kontrol Yapıları:

if-else, switch-case deyimleri, for, while, do-while gibi döngü yapıları, program akışını kontrol etmek için kullanılır.

Fonksiyonlar:

Yeniden kullanılabilir kod parçacıkları oluşturmak için fonksiyonlar tanımlanır. Fonksiyonlar parametre alabilir, değer döndürebilir.

Nesne Yönelimli Programlama:

C++'ın güçlü özelliklerinden biri de nesne yönelimli programlama desteğidir. Sınıflar, nesneler, kalıtım, kapsülleme gibi OOP kavramları derste ayrıntılı şekilde anlatılır.

Giriş/Çıkış İşlemleri:

Klavyeden veri okuma, ekrana yazdırma gibi temel GÇ işlemleri iostream kütüphanesi kullanılarak gerçekleştirilir.

Veri Yapıları:

Dizi, vektör, liste gibi veri yapıları, C++ standart kütüphanesinde yer alır ve programlama örnekleri ile gösterilir.

Ders boyunca, bu kavramların her biri hem teorik hem de pratik açıdan ele alınır. Öğrenciler, çok sayıda örnek kod yazarak, öğrendiklerini pekiştirir.

Dersin sonunda, öğrenciler temel C++ programlama bilgisine sahip olurlar ve daha ileri düzey konulara geçmeye hazır hale gelirler.

C++ intermediate seviye dersi

Bu orta seviye dersi, temel seviye derse kıyasla daha kapsamlı ve gelişmiş konuları ele alır. Ona başlamak için de şuna tıklayın: [Orta Seviye Ders](#)

Bu dersin içerikleri de şöyle:

1. Operatör Yükleme (Operator Overloading):

- Sınıf nesneleri arasındaki matematiksel işlemlerin tanımlanması
- Karşılaştırma operatörlerinin yüklenmesi

2. Kalıtım (Inheritance):

- Temel sınıflardan türetilen alt sınıflar
- Tek ve çoklu kalıtım
- Sanal fonksiyonlar ve sanal tablolar

3. Şablon Programlama (Templates):

- Fonksiyon şablonları

- Sınıf şablonları
- Standart şablon kütüphanesi (STL)

4. Standart Şablon Kütüphanesi (STL):

- Konteynerler (vector, list, set, map vb.)
- İteratörler
- Algoritmalar

5. Dinamik Bellek Yönetimi:

- new ve delete operatörleri
- Akıllı işaretçiler (smart pointers)

6. Dosya İşlemleri:

- Dosya okuma/yazma
- Metin ve ikili dosya işlemleri

7. Özel Durum İşleme (Exception Handling):

- try-catch-throw yapısı
- Özel durum sınıfları oluşturma

8. Makrolar ve Ön İşlemci Komutları:

- #define, #include, #ifdef gibi yapılar
- Conditional derleme

9. Namespaces ve Kapsam Çözümlemesi

10. Çok Biçimlilik (Polymorphism)

11. Sanal Fonksiyonlar ve Soyut Sınıflar

Bu konuların her biri, teorik açıklamalar ve çok sayıda örnek kod parçacıkları ile detaylı şekilde işlenir. Ayrıca, proje tabanlı ödevler ve uygulamalar da yer alır.

Intermediate seviye C++ dersi, öğrencilerin daha karmaşık programlar yazabilme becerilerini geliştirmeyi hedefler. Temel bilgilere hakim olan kişiler, bu dersi tamamladıktan sonra, profesyonel düzeyde C++ programlama yapabilecek yetkinliğe ulaşırlar.

12. Öznitelik Programlaması (Attribute Programming):

- Öznitelik bildirimi ve kullanımı
- `__attribute__` yapısı

13. Gelişmiş Kavramlar:

- Lvalue ve Rvalue kavramları
- Rvalue başvuruları ve `&&` operatörü
- Üyelik fonksiyonlarının `const` bildirilmesi

14. Standart Kütüphane Algoritmaları:

- Sıralama, arama, dönüşüm algoritmaları
- Fonksiyonel programlama yaklaşımları

15. Concurrency ve Eşzamanlılık:

- Çoklu iş parçacıkları (threads)
- Senkronizasyon mekanizmaları
- Atomik işlemler

16. Bellek Yönetimi:

- Statik, otomatik ve dinamik ömür
- Yığın ve yığıt belleği

17. Derleme ve Bağlama:

- Öntanımlı makrolar
- Aşırı yükleme çözümlemesi
- Bağlantı zamanı ve derleme zamanı hataları

18. Derleyici Seçenekleri ve Optimizasyon:

- Derleyici seçeneklerinin etkisi
- Kod optimizasyonu teknikleri

19. Kendi Sınıf Kitaplıklarının Oluşturulması:

- Kapsülleme, soyutlama
- Tasarım desenleri

20. Proje Geliştirme ve Dağıtım:

- Modüler mimari
- Kaynak kodu yönetimi
- Dağıtım ve kurulum

Bu konularla birlikte, derse katılan öğrenciler, profesyonel düzeyde C++ programlama becerilerini geliştirecekler. Ayrıca, kendi sınıf kütüphanelerini tasarlama ve karmaşık projeleri yönetme konularında da deneyim kazanacaklar.

Dersin ilerleyen seviyelerinde, öğrenciler kendi ilgi alanlarına veya ihtiyaçlarına yönelik projeler üzerinde çalışacaklar. Bu sayede, öğrendikleri kavramları ve teknikleri gerçek dünya problemlerini çözmek için kullanma fırsatı bulacaklar.

Bülent abiniz ya da amcanız 😊

Bülent Başaran AFL (87) Bilkent Elektrik ve Elektronik (91) [CMU](#)'da Master (93) ve Doktora (97), [Intel](#) şirketinde yonga tasarımı yüksek mühendisliği sonra [Synopsys](#) şirketinde yonga tasarımı için yazılım tasarım mimarisi, sonra da [Koco'yla](#) Türkçe Yazılım ve Öğrenim Ortamı için gönüllü tasarım.

[Google Gemini Generative AI](#) çok yardımcı oldu bunu yazmakta.