C++ PROGRAMLAMAYA GİRİŞ

DERS DIŞI AKTİVİTE
HAFTA 2

DENE-YAP
YAZILIM TEKNOLOJİLERİ
LİSE

Ders dışı kazanımlar

1. C++ programlama dilinin özelliklerini öğrenir.
2. C++ programlama dili uygulama alanlarını bilir.
3. C++ programlama yapısını bilir.
4. Kod geliştirme ortamı (IDE) ve derleyici kurulumu yapar.

Amaç

Ders dışı aktivitenin amacı, C++ programlama diline giriş yapılarak, dilin geçmişi, kullanım alanları, avantajları ve özelliklerini açıklamak, dile ait temel kavramları öğrenmek kod geliştirme ortamı (IDE) ve derleyici yapılarının çalışma prensibinin incelenmesidir. Öğrenciler Hafta 1 ile Hafta 2 ders saati arasında, ders dışı aktivite olarak gönderilen sunumdaki bilgileri edinip, derleyici kurulumu yapacaktır.

C++ Özellikleri

Sağlam ve verimli programlar tasarlanabilir	Profesyonel programcıların tercihidir	Farklı platformlar için taşınabilir programlar tasarlanabilir
Hızlı ve esnek programlar tasarlanabilir.	Hızlı ve kolay öğrenilir	Diğer programlama dillerinden daha hızlıdır ve mükemmel eşzamanlılık desteği sağlar.
C#, Java ve Python dillerinin atasıdır.	Nesnelerin kullanımını sağlayan popüler bir dildir.	Microsoft Windows, Linux ve Mac OS gibi işletim sistemlerinin yazım dilidir.

C++ Özellikleri

Dili öğrenme ve çalışma kaynağı geniştir.	Dünyaca çalışılan bir dil topluluğu vardır.	C++ öğrenmek için başka bir programlama dili bilmeye gerek yoktur.
Düşük performans ve gecikme istemeyen uygulama ve veri tabanı sunucularının tercihidir.	Yazılımın donanımla yakından eşleştiği alanlarda kullanışlıdır.	Programcı için işleri kolaylaştıran yerleşik bir kütüphaneye sahiptir.

C++ Özellikleri

Çeşitli ortamlarda aynı kod parçasının kullanılmasını sağlar.	Basit içeriği ile yeni programcıların gözdesidir.	Taşınabilirlik özelliği sayesinde çeşitli ortamlarda aynı kod parçasının kullanılmasını sağlar.
Yüksek seviyeli dili ile belleği dinamik olarak yönetmeye izin verir.	Java ve Python'dan farklı olarak yorumlayıcı tabanlı değil derleyici tabanlı bir dildir.	

C++ Uygulama Alanları

Yeni İşletim Sistemi Geliştirme: *Microsoft Windows, Linux ve Mac OS gibi işletim sistemleri C++ ile programlanmıştır. Özellikle hızlı bir programlama dili olması nedeniyle işletim sistemlerinin temelidir ve işletim sistemi geliştirmek için ideal bir seçimdir. Buna ek olarak, C++, düşük düzeyli programlar yazmaya yardımcı olan geniş bir sistem düzeyi işlevleri koleksiyonuna sahiptir.*

Derleyici Geliştirme: *Bazı programlama dillerinin derleyicileri arka uç (backend) programlama dili olarak C++ kullanmaktadır. Bunun nedeni, C++ dilinin diğer dillere göre daha düşük seviyeli ve donanıma daha yakın bir dil olmasıdır. Ayrıca, performansı korurken, mantıksal ve nesne yönelimli bir şekilde geliştirmeye izin veren üst düzey bir özelliğe sahiptir. İlk C++ derleyicisi olan Cfront, C++ ile yazılmıştır.*

Gömülü Sistemler: *Robotik, tıbbi ekipman sistemleri ve akıllı cihazlar gibi çeşitli gömülü sistemler, C++ dilinin diğer yüksek seviye programlama dillerine kıyasla donanım seviyesine daha yakın bir dil olması nedeniyle uygulamalarda kullanılabilir. C++, diğer yüksek seviye programlama dilleriyle karşılaştırıldığında çok sayıda düşük seviyeli işlev çağrısı sağlayabilir.*

C++ Uygulama Alanları

Web Tarayıcı Oluşturma: *Çeşitli web tarayıcılarının görselleştirme motorları C++ dilinde programlanmıştır. Burada ön plana çıkan da dilin sunduğu hızdır. Kullanıcıların içeriğin ekranda görüntülenmesini uzun süre beklememesi için daha hızlı çalışması gerekir. Dolayısıyla, bu tür düşük gecikmeli olması gereken sistemler programlama dili olarak C++ kullanır.*

Oyun Programlama: *Tüm grafik uygulamaları hızlı işleme gerektirir ve oyun programlamada da grafik işlemleri ön planda olduğu için C++ gecikmeyi azaltmaya yardımcı olur. Grafik işlemleri yoğun popüler oyunlar bile birincil dili olarak C++ kullanır. Dilin sağladığı hız, geliştiricilerin hedef kitlelerini genişletmelerine yardımcı olur. Optimize edilmiş bir uygulama, yüksek hesaplama gücüne sahip olmayan düşük seviye cihazlarda bile çalışabilir.*

C++ Uygulama Alanları

Masaüstü (Hesaplama) Uygulamaları: *C++, kullanıcı arayüz tabanlı ve masaüstü uygulamalarının çoğunu gerekli özelliklere sahip olduğu için kolayca geliştirmek için kullanılabilir. Örneğin; Adobe sistemlerinin çoğu uygulaması C++ kullanılarak geliştirilmiştir. Temel bankacılık uygulamaları milyonlarca işlemi günlük olarak işler ve yüksek eşzamanlılık ve düşük gecikme süresi desteği gerektirdiği için arka uç programlama dili olarak C++ kullanılırlar.*

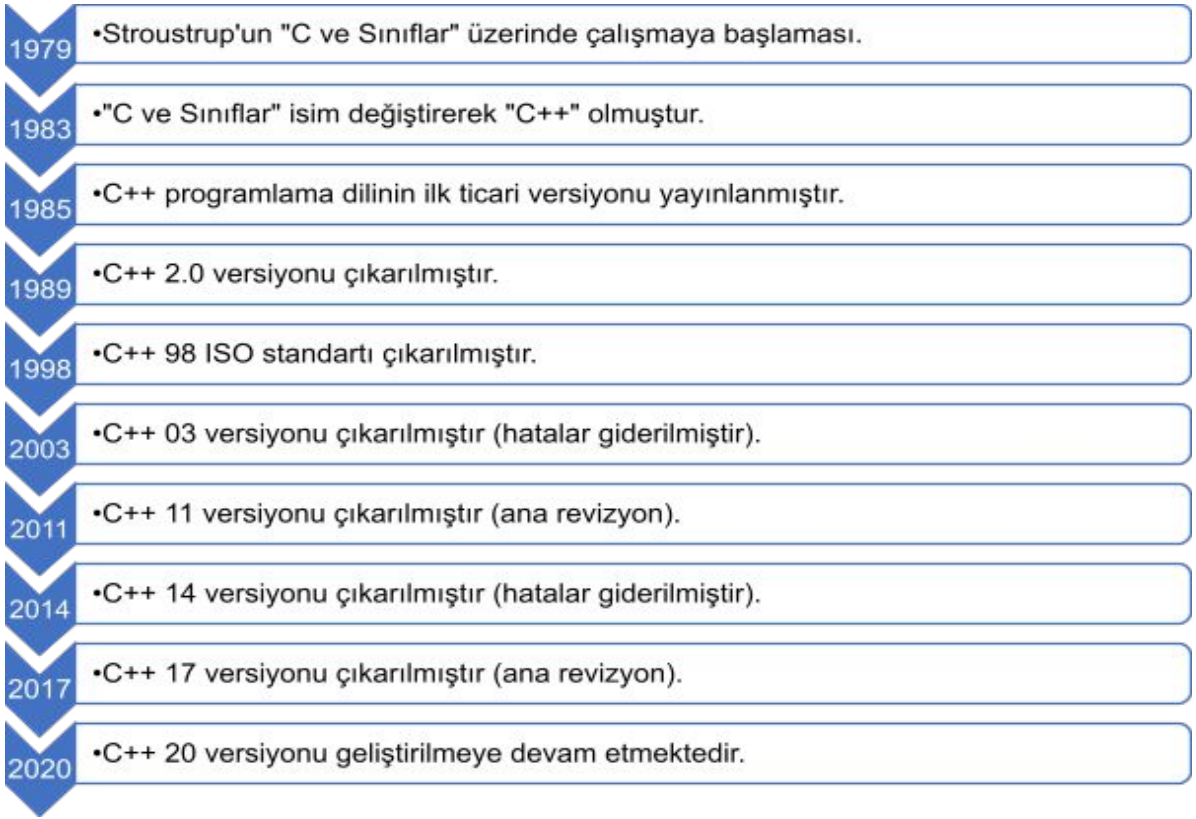
Yeni Programlama Dili Geliştirme: *C++ kullanarak kendinize yeni bir dil icat edebilirsiniz. Kod değiştirme işlemi olarak yeni dilin çalışmasını sağlayabilirsiniz. Çoğu yeni programlama dili, dilin ayrıştırıcısını oluşturmak ve işletim sisteminin çalışma zamanını yönetmek için bir C++ arka ucuna sahiptir.*

C++ Uygulama Alanları

Unutmayın ki bir programlama dili ile yapabileceklerinizin sınırı yoktur. Sadece bazı diller aynı uygulamayı yapmanızı daha kolaylaştırır.

YouTube, Google, Amazon, Twitter ve Facebook gibi uygulamaların yapımında C++ programlama dili de kullanılmıştır. Dolayısıyla dünya çapında popülerliğini korumaktadır. Popüler programlama topluluklarının yayınlamış olduğu indekslere göre de C++ hala kullanılan bir programlama dilidir.

C++ dilinin geçmişi



C++ programlama yapısı

Adım 1: Problemi tanımlayın
Çözmek istediğiniz problemi belirlediğiniz ilk adımdır. Programlamak istediğimiz problemi en iyi şekilde tanımlayarak bir sonraki adıma geçebiliriz. Örnek olarak; “klavyeden girilen 10 tane sayının ortalamasının hesaplanması”, “verilen matris üzerindeki iki nokta arasındaki en kısa yolun bulunması” ya da “market stoklarındaki bir ürünün miktarının stok dosyasından kontrol edilmesi” olabilir.

Adım 2: Çözümü tasarlayın
Birinci adımda belirlediğimiz problemin nasıl çözüleceğini belirlediğimiz adımdır. Problemin çözümünde birden fazla yol olabileceği için bu çözümlerin detaylı bir şekilde analiz edilmesi gerekir. İlk akla gelen çözümle bir sonraki adıma geçilmemelidir. En basit ve hızlı çözüm tespit edilmeli ve bu çözümü uygulayan programın yazımına geçilmelidir.

C++ programlama yapısı

Adım 3: Çözümü uygulayan programı yazın
Programı yazmak için öncelikle hangi programlama dilini kullanacağımıza karar vermemiz gerekir. Daha sonra da bu dile uygun bir editöre ihtiyacımız olacaktır. Kodlarımızı basitçe not defterine de yazabiliriz fakat kodlama için hazırlanmış editörleri tercih etmemiz gerekir. Sonraki bölümde kod düzenleyici editörlerden bahsedeceğiz. Programımızı yazdıktan sonra, bir sonraki adımda programımızı çalıştırabileceğimiz bir şeye dönüştürmemiz gerekecektir.

Adım 4: Programı derleyin
Bir önceki adımda yazdığımız programı derlemek için C++ derleyicisi kullanıyoruz. C++ derleyicisi, programınız için hazırladığınız kaynak kodu dosyasını sırasıyla satır satır inceler. Bu incelemede yazılan kodun C++ dilinin kurallarına uygunluğunu kontrol eder. Eğer kodunuzda bir hata varsa ise derleyici ilgili satır numarasını belirterek neyin düzeltilmesini gerektiğine yardımcı bir hata bilgisi verir ve derleme işlemini sonlandırır. İlgili hatalar giderildikten sonra C++ kaynak kodumuz nesne dosyası adı verilen bir makine dili dosyasına çevrilir.

C++ programlama yapısı

Adım 5: Nesne dosyalarını bağlayın
Derleyici nesne dosyasını oluşturduktan sonra, bağlayıcı adı verilen başka bir program devreye girer. Bağlayıcı, derleyici tarafından oluşturulan tüm nesne dosyalarını tek bir yürütülebilir programda birleştirir. Bağlayıcı nesne dosyalarının yanı sıra kütüphane dosyalarını da bağlayabilir. Kütüphane dosyası, önceden derlenmiş kod koleksiyonudur. Bağlayıcı tüm nesne dosyalarını ve kütüphaneleri bağlamayı bitirdiğinde çalıştırabilir bir dosya oluşturur.

Adım 6: Programı test edin
Sürecin en kolay ve eğlenceli kısmıdır. Yürütülebilir dosyayı çalıştırabilir ve beklediğiniz çıktıyı üretip üretmediğini kontrol edebilirsiniz. Eğer her şey istediğiniz gibi çalışıyorsa programınızı kullanıcılar ile paylaşabilirsiniz. Programınız çalışıyor ancak beklediğiniz sonuçları üretmiyorsa hata ayıklama (debug) adımına geçmeniz ve programınızı adım adım test etmeniz gerekir.

C++ programlama yapısı

Adım 7: Debug

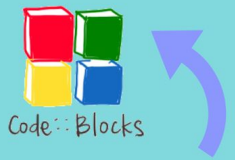
Programımızdaki beklenmedik sonuçları çözmek için hata ayıklama yani debug adımına geçeriz. Bu aşamada yazdığımız kodun her bir satırında yapılan işlemleri tek tek kontrol ederek mantıksal hatanın giderilmesini sağlarız. **Yapılan araştırmalara göre programlamada zamanının sadece %20'lik kısmı programı yazmak için harcanırken, %80'lik kısmı hata ayıklamada yani hataları gidermede harcanmaktadır. Buradan, ikinci adım olan çözümü tasarlamaya daha fazla önem verilmesi gerektiği ortaya çıkmaktadır.**

DERLEYİCİ KAVRAMI

Bir
bilgisayarı doğrudan makine dilinde yani yalnızca **sıfır** ve **bir** rakamını kullanarak herhangi bir program yazabilir miyim acaba?

Bilgisayara istediğim talimatları vermek için
binlerce, belki milyarlarca sıfır ve bir rakamını bir araya getirmek zor olsa gerek!!!

Keşke
benim bilgisayara vermek istediğim talimatlarımı 0 ve 1 rakamına dönüştürecek bir şeyler olsaydı?



Yaşasın derleyiciler bunun içinmiş!!!

C++
derlenmiş bir dil olarak tasarlandığı için sistem tarafından doğrudan makine diline çevrilir ve bu da oluşturulan programı oldukça verimli hale getirir.

-HADİ O ZAMAN DERLEYİCİMİZİ SEÇELİM-

KODLAMA YAPMAK İÇİN HANGİ DERLEYİCİYİ SEÇMELİYİM?

- Seçtiğim derleyici;
  - Bilgisayarımdaki işletim sistemime uygun mu?
  - Ücretli mi yoksa açık kaynak kodlu mu?
  - Performans açısından verimli mi?
  - Sonucu hızlı bir şekilde üretiyor mu?
  - Tüm kod kitaplıklarını ve araçlarını içeriyor mu?

Bu eğitim kapsamında kodlarımızı derlemek için her platform tarafından desteklenen ve rahat kullanıma sahip ***Code::Blocks*** geliştirme ortamı tercih edilmiştir.

Derleyiciyi Kuralım!!

1. Öğrenme Yönetim Sistemi üzerinde Derleyici kurulum kılavuzunu açınız.

2. Kılavuz adım adım kurulumda ilerlemeyi gösterir. Kılavuzdaki her bir adımı aşamalar halinde tamamlayıp, derleyiciyi bilgisayarınıza kurunuz.

3. Kurulum kılavuzuna ayrıca buradan erişebilirsiniz.