

**GRUP 5**  
**PROGRAMLAMA DİLLERİNİN KAVRAMI FİNAL PROJE**  
**ÖDEVİ C++ PROGRAMLAMA DİLİ RAPORU**

Hazırlayanlar:

21100011010 Metin Furkan YAMAN

21100011050 Sema Nur EKMEKÇİ

# C++ Nedir?

C++, Danimarkalı bilgisayar bilimci Bjarne Stroustrup tarafından geliştirilmiş bir dil olup, nesne yönelimli bir programlama dilidir ve düşük seviye ile yüksek seviye dilin bir kombinasyonunu içerir. Bu özellikleriyle C++, orta seviyeli bir programlama dilidir.

## Neden C++ Öğrenmeliyiz?

- C++ en çok kullanılan ve popüler programlama dillerinden biridir.
- C++ işletim sistemleri, gömülü sistemler ve Grafiksel Kullanıcı Arayüzlerinin yapımında kullanılır.
- Soyutlama (Abstraction), Kapsülleme (Encapsulation), “Kalıtım (Inheritance) ve Çok Biçimlilik (Polymorphism) olan tüm OOP kavramlarını uygulayan, programlara net bir yapı kazandıran ve kodun yeniden kullanılmasına olanak tanıyan, geliştirme maliyetlerini düşüren ve güvenlik sağlayan nesne yönelimli bir programlama dilidir.
- Taşınabilirlik açısından esnektir ve birden fazla platforma (cross platform) uyarlanabilen uygulamalar oluşturmak için kullanılabilir.
- C++'ın öğrenilmesi kolaydır.
- Söz dizimi (Syntax) C, Java ve C#'a benzerdir.



*Resim 1. C++ Resmi Olmayan Sembolü*

# 1. Tarihsel Geçmiş

Bjarne Stroustrup, nesne yönelimli programlamayı destekleyen ilk dil olarak kabul edilen Simula dilinde çalışmalar gerçekleştirmiştir. Simula'nın yazılım geliştirmek için oldukça faydalı ancak pratik kullanım için yavaş olduğunu fark etmiştir. Bu farkındalık, Stroustrup'u C dilini genişleterek nesne yönelimli programlamayı destekleyen bir yapı oluşturma fikriyle yeni bir dil üzerinde çalışmaya itmiştir. Bu doğrultuda "C with classes (Sınıflarla C)" konseptiyle yeni bir dil geliştirme çalışmalarına başlamıştır. "C with classes" diline derleyici geliştirmek için C dilinin derleyicisi olan Cpre'den türetilen Cfront derleyicisini yazmıştır. Cfront, C with Classes kodunu normal C'ye çevirmek için tasarlanmış bir programdı. İlginç bir nokta ise Cfront'un büyük bir kısmının C with Classes ile yazılmış olmasıydı; bu da onu kendi kendini derleyebilen (bir derleyicinin kendi kodunu derleyebilme yeteneği) bir derleyici haline getiriyordu[1]. Cfront, yeni özelliklerin (C++ istisnalar(try-except)) entegre edilmesinin zorlaşmasının ardından 1993 yılında kullanımı sonlandırılmıştır. Cfront, her ne kadar bazı eksikliklere sahip olsa da gelecekteki derleyicilerin geliştirilmesine ve Unix işletim sistemi üzerindeki etkisine önemli bir katkısı olmuştur.

## 1.1 C++ (1983)

1983 yılında dilin adı "C with Classes"tan C++ olarak değiştirilmiştir. C++'da "++" operatörü bir değişkeni arttırmak amacıyla kullanılır, bu operatörün dilin isminin C++ olarak değiştirilmesinde etkisi vardır. Bu süre zarfında gelen güncelleme ile bir takım yeni özellikler eklenmiştir[1]. Bunlar arasında en dikkate değer olanlar:

- Virtual Functions (Polymorphism)
- Function Overloading
- Referans Sembolü (&)
- Const anahtar sözcüğü
- İki eğik çizgi ile yorum satırıdır.

## 1.2 C++ (1985)

Ticari bir dil olarak 1985 yılında piyasa sürülmüştür. "The C++ programming language" adıyla Bjarne Stroustrup tarafından referans yayınlandı. Dil henüz resmi olarak standartlaştırılmamış olması kitabı çok önemli bir referans haline getirmiştir.

## 1.3 C++ 2.0

1989 yılında C++'ın ilk resmi standardı olan C++ 2.0 versiyonu yayınlanmıştır. 2.0 sürümü öncesinde yayınlanan C++ sürümleri resmi standartlaştırma olmadığı için farklı derleyiciler arasında uyumluluk sorunlarına sebep oluyordu. C++ 2.0 ile bu sorunların çözülmesine ve C++'ın popülerliğini artırarak daha yaygın kullanılan bir dil haline getirmiştir [2].

## 1.4 C++ (1990)

1990 yılında “The Annotated C++ manuel” referans kılavuzu yayınlanmıştır. Aynı yıl Borland ‘ın Turbo C++ derleyicisi de piyasaya sürülmüştür. Turbo C++ derleyicisi , C++ dilinin gelişimi üzerinde önemli etkiye sahip olacak çok sayıda ek kütüphane eklemiştir. Turbo C++’ın son kararlı sürümü (stable release) 2006 yılında yayınlanmasına rağmen derleyici hala yaygın olarak kullanılmaktadır [1] .

## 1.5 C++ (1998)

1998 yılında, C++ Geleneksel Danışma ve Standart Komitesi, C++ için ilk uluslararası standart olan ISO/IEC 14882:1998’i yayınladı. Standart Şablon Kütüphanesi de bu yıl dahil edilmiştir. 2003 yılında, komite, 1998 standartlarına uyumlu olan ve bir dizi sorunu düzelten bir güncelleme gerçekleştirdi. Değiştirilmiş dil, C++03 olarak adlandırıldı [1].

## 1.6 C++05

2005 yılında, C++ komitesi, en yeni C++ standardını geliştirmeyi amaçlayan çeşitli seçenekleri detaylandıran bir teknik rapor (TR1 olarak adlandırılan) yayımladı [1].

## 1.7 C++11

2011 yılında yayınlanan bu sürümde Boost kütüphane projesi, yeni standarda önemli bir etki yaptı ve bazı yeni modüller doğrudan ilgili Boost kütüphanelerinden türetilmiştir [2]. Eklenen bazı yeni özellikler şunlardır:

- Belirli diğer dillerdeki foreach döngülerine benzer işlevselliği sunan yeni for döngüsü sözdizimi
- Standart çoklu iş parçacıklı kütüphane (2011 yılından itibaren eklenmiştir.)
- AUTO anahtar kelimesi
- Yeni C++ zaman kütüphanesi, atomik destek
- Kapsamlı koleksiyon kütüphanesi
- Düzenli ifade desteği
- Birlik ve dizi başlatma listeleri için iyileştirilmiş destek

## 1.8 C++14

2014 yılında yayınlanan bu sürümde öne çıkan özellikler şunlardır [1]:

- Basamak Ayırıcılar
- Genelleştirilmiş Lambda Yakalama:
- Değişken Şablonları
- İkili Tam Sayı Literalleri
- Quoted strings
- Polimorfik Lambdalar

## 1.9 C++17

2017 yılında yayınlanan bu sürümde öne çıkan özellikler şunlardır [2]:

- Fold ifadeleri
- Onaltılık kayan nokta literalleri
- u8 karakter literal
- Seçim ifadeleri ile başlatıcı
- Satır içi değişkenler

## 1.10 C++20

2020 yılında yayınlanan bu güncelleme ile C++:

- Değişkenler,
- Enumarationlar,
- Classlar ve üyeleri,
- Lambdalar ve yakalamaları gibi program öğelerini inceleme özellikleri kazanmıştır [1].

## 1.11 C++23

C++23 ile yeni eklenen özellikler arasında örtülü hareketin basitleştirilmesi, aralık döngülerinde geçicilerin sabitlenmesi, çok boyutlu ve statik operator ve Unicode iyileştirmeleri yer almaktadır. Ayrıca constexpr fonksiyonlarda static constexpr değişkenler tanımlama öne çıkar [1].

# 2. Tasarım Süreci

## 2.1 C'den C++'a Tasarım Süreci

Bjarne Stroustrup, C++'ı tasarlarken, C dilinin özelliklerini ve sınırlamalarını detaylı bir şekilde incelemiştir.

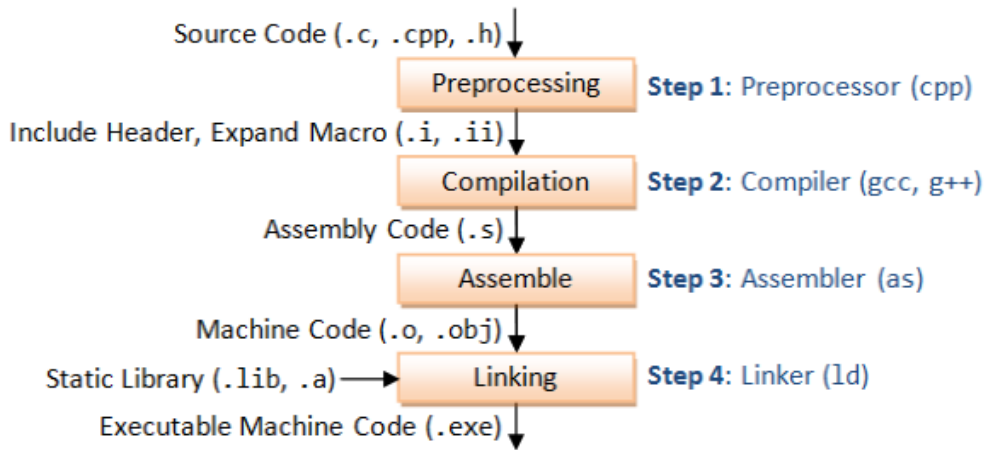
C++, genel amaçlı bir programlama dilidir ve temel olarak C dilinin nesne yönelimli bir yapı içerecek şekilde geliştirilmiş bir versiyonudur. C++'ın geliştirilmesindeki amaç, C dilinin zayıf yönlerini geliştirmek ve nesne yönelimli programlama prensiplerini ele alarak dilin kapsamlılığını artırmak ve tür güvenliğini ve hata yakalama mekanizmalarını iyileştirmektir. C++'ın öne çıkan özelliklerinden biri, nesne yönelimli programlamayı desteklemesidir. C++ kullanıcılara, programlama sırasında nesneler oluşturma ve yok etme yeteneği sağlar. Bu, C++'ın prosedürel bir programlama dili olan C'den farklılaştığını gösterir.

C++ ayrıca, nesne yönelimli programlama yeteneklerinin yanı sıra yapısal ve disiplinli programlamaya olanak veren özellikler sağlar. Bu özellikler, programcılara kodlarını modülerleştirmeleri, daha iyi organizasyon sağlamaları ve daha sürdürülebilir yazılım geliştirmeleri için araçlar sunar. C++ Standart Kütüphanesi, tüm programcıların

kullanabileceği zengin bir sınıf ve fonksiyon koleksiyonu sunar. Bu sayede, büyük ve karmaşık projelerde çalışırken daha etkili bir kod yazma deneyimi sunar. Sonuç olarak, C++ hem düşük seviyeli, hızlı ve etkili C dilinin mirasını devralırken, aynı zamanda nesne yönelimli programlama prensiplerini içeren bir dil haline gelmiştir .

## 2.2 C++'da Derleme Süreci

C++'da Sınıflar (varsa) başlıklara ( \*.h uzantılı dosyalar , sınıfın bildirimi) ve uygulamalara (.cpp uzantılı dosyalar, sınıfın tanımı) ayrılır. Her sınıf ayrı bir derleme birimi (compilation unit) oluşturur . Proje oluşturduğunda kaynağı ikili koda çeviren birkaç temel bileşen vardır [3].



Resim 2. C++ Derleme Süreci  
Şeması

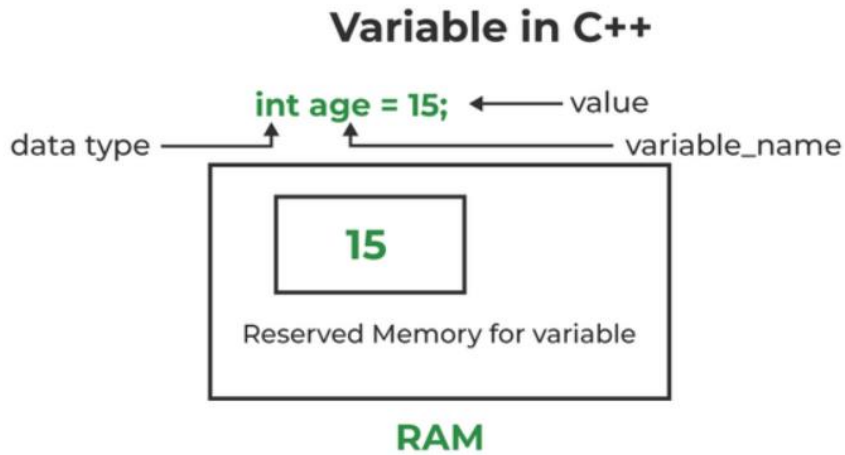
İlk olarak , başlık dosyalarının dahil edilmesinden ve sunulduğu takdirde makro ifadelerin çevirisinden sorumlu olan ön işlemci (preprocessor) gelir .

Ardından derleyiciler(compiler) kodu makine diline çevirir ve assembly kodu olan “.s” uzantılı dosyası oluşturur.

Bundan sonra assembler, assembly kodunu nesne koduna (.o ve .obj uzantılı dosyalar, makine komutları dönüştürür.

Son olarak, bağlayıcı (linker) tüm nesne dosyalarını alır ve bunları diğer benzer dosyalarla birleştirmenin yanı sıra yürütülebilir bir dosya veya kitaplık oluşturur.

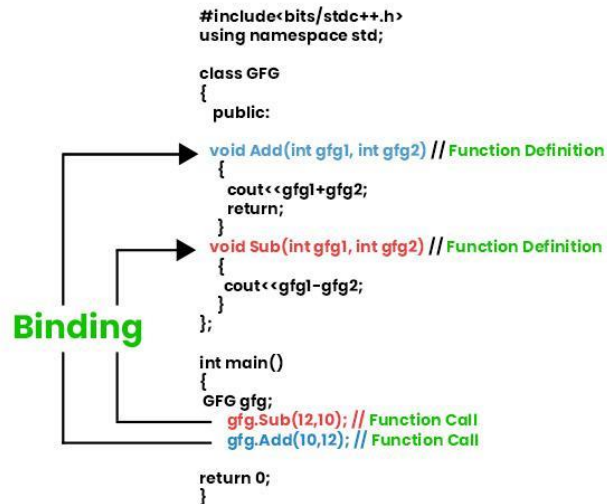
## 2.3 C++'da Tip Bağlama Süreci



Resim 3. C++ Değişken Tanımlama

Resim 3'te görsel C++'da bir değişkenin nasıl tanımlandığını göstermektedir. Değişkenin adı "age" ve türü "int"tir. Değişkenin değeri 15'tir. Değişken, RAM'de ayrılmış bir bellek alanına bağlanır. RAM'de 4 baytlık bir bellek alanı ayrılır. Bu alan, değişkenin değeri olan 15'i depolamak için kullanılır. Bu süreç, derleyici tarafından gerçekleştirilir [4].

### 2.3.1 Dinamik Bağlama



Resim 4. C++ Dinamik Bağlama

C++'daki dinamik bağlama, derleme zamanında meydana gelen statik bağlamayla ilgili sorunlardan kaçınarak işlev çağrılarını işlev tanımlarına bağlama uygulamasıdır. Dinamik bağlama esnek olduğundan, işlev çağrısını ve tanımı derleme sırasında birbirine bağlayan statik bağlamanın dezavantajlarını ortadan kaldırır [12].

## 2.3.2 Statik Tip Bağlama

C++'da statik bağlama (erken bağlama veya derleme zamanı bağlama olarak da bilinir), bir işlev çağrısının çalışma zamanından ziyade derleme zamanında çözülmesini ifade eder. Derleyici, kullanılan nesnenin statik (bildirilen) türüne göre tam olarak hangi işlevin çağrılacağını bildiğinde ortaya çıkar [13].

```
1  #include <iostream>
2
3  class Hayvan {
4  public:
5      void sesCikar() {
6          std::cout << "Hayvan: Ses Çıkardı\n";
7      }
8  };
9
10 int main() {
11     Hayvan h;
12
13
14     h.show(); // Hayvan classının sesCikar() metodu çağrılır (statik bağlama)
15
16     return 0;
17 }
18
```

Resim 5. C++ Statik Tip Bağlama

# 3. Dilin Genel Görünümü ve Değerlendirilmesi

## 3.1 C++ Dili Özelinde Değerlendirme Kriterleri

### 3.1.1 Okunabilirlik (Readability)

Programlama dillerinde okunabilirlik, bir programın ne kadar anlaşılır ve düzenli olduğunu ifade eder. Okunabilir kod, hem yazılım geliştiricilerin hem de diğer kişilerin yazılı kodu daha hızlı ve etkili bir şekilde anlamalarını sağlar.

Characteristic	CRITERIA		
	READABILITY	WRITABILITY	RELIABILITY
Simplicity	•	•	•
Orthogonality	•	•	•
Data types	•	•	•
Syntax design	•	•	•
Support for abstraction		•	•
Expressivity		•	•
Type checking			•
Exception handling			•
Restricted aliasing			•

Resim 6. Dil Değerlendirme Kriterleri ve Bunları Etkileyen Özellikler [14]



- C++ dilinin nesne yönelimli bir dil olması ile modüler yapının oluşması okunabilirliği olumlu etkiler.
- Ortogonalite okunabilirliği etkileyen diğer faktörlerden biridir. Ortogonalite, dilin özelliklerinin birbirleriyle bağımsız olması demektir. Ortogonalite mantıksal olarak "A'yı değiştirmek B'yi değiştirmez" olarak tanımlayabiliriz. Bu, dilin özelliklerinin birbirine mümkün olduğunca az bağımlı olması anlamına gelir. Okunabilirlik açısından, ortogonalite genellikle avantaj sağlar çünkü bu özelliklerin birbirlerinden bağımsız olması, kodun daha anlaşılır ve tahmin edilebilir olmasını sağlayabilir. C++ tamamen ortogonal bir dil değildir. Örneğin, C++'daki bir fonksiyondan bir yapıyı (struct) döndürebiliriz, ancak bir diziye döndüremeyiz [5].
- Operatör özellik çokluğu okunurluğu olumsuz etkileyen faktörlerden biridir. Örneğin: C++ dilinde "+" ve "\*" operatörleri, birden fazla farklı işlem yapmak için kullanılabilir.

```

main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int sayi = 0;
6      // 1. Yöntem: sayi = sayi + 1;
7      sayi = sayi + 1;
8      // 2. Yöntem: sayi += 1;
9      sayi += 1;
10     // 3. Yöntem: sayi++;
11     sayi++;
12     // 4. Yöntem: ++sayi;
13     ++sayi;
14
15     cout << "Artirilmis sayi degeri: " << sayi << endl;
16
17     return 0;
18 }

```

Artirilmis sayi degeri: 4

Process returned 0 (0x0) execution time : 0.014 s  
Press any key to continue.

**Resim 7. C++ Operatör Özellik Çokluğu**

Resim 7 'de örnekte 4 kod satırı da sayıyı 1 artırır.

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 5;
6      int* ptr = &x;
7      int sonuc = x * 2;
8
9      cout << "ptr'nin gosterdigi deger: " << *ptr << endl;
10     cout << "Carpim sonucu: " << sonuc << endl;
11
12     return 0;
13 }

```

ptr'nin gosterdigi deger: 5  
Carpim sonucu: 10

Process returned 0 (0x0) execution time : 0.011 s  
Press any key to continue.

**Resim 8. C++ Operatör Özellik Çokluğu**

Resim 8 'de "\*" operatörü 6. satırda oluşturulan ptr işaretçisi (pointer) x değişkeninin adresini tutmaktadır. Aynı zamanda 7. satırda bulunan x değişkeni çarpma işleminde kullanılmaktadır.

### 3.1.2 Yazılabilirlik (Writability)

Yazılabilirlik, bir yazılımın kolayca anlaşılabilir, düzenlenebilir ve sürdürülebilir olma özelliğidir. Yazılabilirliği yüksek olan dillerde basitlik, soyutlama desteği, ortogonalite durumu, anlatımcılık durumu öne çıkmaktadır.

- Bir dilin yazılabilir olması demek kullanılacak dilin projeye uygun olması demektir. Örneğin; C++ dilinde gömülü sistem ve web projelerinin yazılabilirliğini karşılaştırmak doğru değildir. C++ dili gömülü sistem projelerinde kullanmaya daha elverişli bir yapıya sahiptir. C++ dilinde oyun projeleri, gömülü sistem projeleri, bulut/dağıtık sistem projeleri, GUI tabanlı uygulamaların yazılması yazılabilirliği yüksek projelerdir.
- C++, nesne yönelimli programlama prosedürünü desteklediği için modüler bir yapıya sahiptir ve bu da yazılabilirliği artırır. Bu aynı zamanda hata kontrolünü de kolaylaştırır.
- C++'ın standart kütüphanesi, pek çok fonksiyonu hazır olarak sunar. Bu kütüphane, tekrar kullanılabilir ve güvenilir kod yazmada yardımcı olur.

### 3.1.3 Güvenilirlik (Reliability)

Bir program tüm şartlar altında özelliklerini gerçekleştirebiliyorsa güvenilirdir.

- C++ dili derlenen bir dildir. Bundan dolayı tip kontrolü statik yapıya sahiptir. Statik Tip Kontrolü, program derleme zamanında gerçekleştirilir. Derleme esnasında tip kontrolleri yapıldığı için programın çalışması sırasında, ekstra tip kontrolü yapılmaz. Bu da uygulamaların çok daha verimli, hızlı ve güvenli çalıştığı anlamına gelebilir. Derleyici, derleme esnasında herhangi bir hata yakalamadı ise bir tip hatasının oluşmadığı kesindir.
- C++ bellek yönetimi konusunda esnek bir yapıya sahiptir. Bu özelliği ile birlikte güvenilirliği olumsuz etkileyebilir. Örneğin ayrılan belleği geri vermemek ve bellek kaynaklarını serbest bırakmamak gibi durumlar, bellek sızıntılarına yol açabilir. Bellek alanı serbest bırakıldıktan sonra o alana işaret eden pointerlar sallanan işaretçiler (dangling pointer) durumuna sebebiyet verebilir ve bu da programın hatalı çalışmasına/çökmesine sebep olabilir.

### 3.1.4 Maliyet

Programlama dillerinde maliyet, genellikle kaynak kullanımı, zaman, bellek kullanımı, personel eğitimi, lisans ücreti veya yazılım geliştirme sürecinde harcanan emek gibi faktörlerle ilişkilidir.

- C++ açık kaynak olmakla beraber lisans ücreti istemeyen bir yazılım dilidir.
- C++ dilinin maliyeti, sunduğu kontrol ve esneklikle birlikte gelir. Doğru kullanıldığında, yüksek performanslı ve optimize edilmiş yazılımlar geliştirmeye imkan tanır. Ancak, bu avantajlar yanlış uygulandığında veya dikkatlice ele alınmadığında, maliyetli hatalara ve bakım zorluklarına yol açabilir.

## 3.2 C++'ın Diğer Dillere Göre Avantajları

### 3.2.1. Nesneye Dayalı

C++ nesne yönelimli bir programlama dilidir ; bu, ana odak noktasının nesneler ve bu nesnelerin etrafındaki manipülasyonlar olduğu anlamına gelir. Bu, bir dizi hesaplama adımının gerçekleştirilmesini gerektiren prosedürel veya yapısal programlamanın aksine, kodun işlenmesini çok daha kolaylaştırır.

### 3.2.2. Hız

C++, hızın kritik bir ölçüm olduğu durumlarda sıkça tercih edilen bir dil olarak öne çıkar. Derleme ve yürütme süreleri, genel amaçlı programlama dillerine kıyasla belirgin bir hız avantajı sunar.

### 3.2.3. Derlenmiş

Derleme gerektirmeyen diğer programlama dillerinden farklı olarak, her C++ kodunun önce düşük seviyeli bir dile derlenmesi ve ardından çalıştırılması gerekir.

### 3.2.4. Zengin Kütüphane Desteği

C++ Standart Şablon Kitaplığı'nda (STL), kodun hızlı bir şekilde yazılmasına yardımcı olacak birçok işlev bulunur. Örneğin karma tabloları, haritalar, kümeler vb. gibi çeşitli kapsayıcılar için STL'ler vardır [6].

### 3.2.5. İşaretçi Desteği

C++ ayrıca diğer programlama dillerinde sıklıkla bulunmayan işaretçileri de destekler [6].

### 3.2.6. Donanıma Daha Yakın

C++, genel amaçlı programlama dillerine kıyasla donanıma daha yakın bir konumda bulunur. Bu durum, donanım ve yazılım arasındaki sıkı bağlantıyı ifade eder ve yazılım düzeyinde düşük seviyeli destek gerektiren alanlarda özellikle faydalıdır.

## 4. Kökeni ve Karakteristikleri

### 4.1 C++ dinamik bellek kullanımı

Dinamik olarak bellek tahsisi program çalışırken heap adı verilen bellek bölgesinden yapılır. C++'da dinamik bellek işlemleri için new ve delete işlemcileri kullanılır. New işlemcisi bellek tahsisi yapar ve tahsis edilen belleğin başlangıç adresini geri döndürür. Delete işlemcisi daha önce new işlemcisi ile tahsis edilen belleği boşaltır.

```

1  #include <iostream>
2  #include <stdlib.h> // exit() fonksiyonu için
3
4  // veri-türü işaretci-adı = new veri-türü;
5  |
6  // delete işaretci-adı;
7
8  using namespace std;
9
10 int main(void)
11 {
12     int *ip;
13
14     ip = new int; // int değer huyukluğunda bellek tahsis eder.
15
16     if (!ip) {
17         cout << "Bellek tahsis hatası!";
18         exit(1); // Programı sona erdirir.
19     }
20
21     *ip = 21; // işaretci yoluyla bellekte değer atar.
22
23     cout << ip << " bellek adresindeki değer: " << *ip;
24
25     delete ip;
26
27     return 0;
28 }

```

**Resim 9. C++ Bellek Kullanımı**

## 4.2 C++'da Nesne Yönelimli Programlama (Object Oriented Programming)

Nesne Yönelimli Programlama (NYP) mantıksal işlemlerden ziyade, nesnelere (object) ve nesneler üzerinde işlemlere odaklanan programlama dili modelidir. NYP'de programlar, nesnelerin birbirleriyle etkileşime geçmeleri sağlanmasıyla tasarlanır. C++'da nesne yönelimli programlama dillerinden biridir. NYP teorisinde 4 temel özelliğin gerçekleştirilmesi zorunlu sayılmıştır ve biri bile eksik ise bu dil saf NYP sayılmamıştır. Bunlar:

### 4.2.1 Kapsülleme (Encapsulation)

Sınıfta tanımlanan öğeye kimlerin erişebileceğini belirtme olanağı tanır. Bunlara erişim belirteçleri (Access modifiers, access levels) denir. Erişim belirteçleri şunlardır:

Erişim Belirteci	İzinler
public	Bütün sınıflar erişebilir
private	Alt-sınıf dahil başka hiçbir sınıf erişemez
protected	Alt-sınıflar ve aynı paketkiler erişebilir
<belirteç yok> (ön tanımlı, friendly)	Aynı paketkiler erişebilir

**Resim 10. C++ Erişim Belirteçleri**

### Kapsülleme kod örneği:

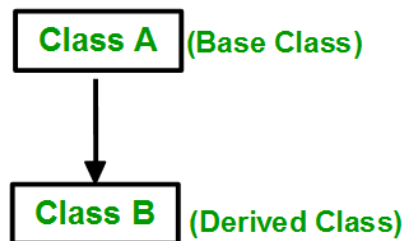
```
1  #include <iostream>
2
3  class Sınıfım {
4  public: // Genel erişim belirtici (Bütün Sınıflar Erişebilir)
5      int publicVar; // Genel erişimli bir değişken
6      void publicFunc() {
7          std::cout << "Bu fonksiyon genel erişimlidir." << std::endl;
8      }
9
10 private: // Özel erişim belirtici (Alt-sınıf dahil başka hiçbir sınıf erişemez)
11     int privateVar; // Özel erişimli bir değişken
12     void privateFunc() {
13         std::cout << "Bu fonksiyon özel erişimlidir." << std::endl;
14     }
15
16 protected: // Korunmalı erişim belirtici Alt-sınıflar ve aynı pakettekiler erişebilir.
17     int protectedVar; // Korunmalı erişimli bir değişken
18     void protectedFunc() {
19         std::cout << "Bu fonksiyon korunmalı erişimlidir." << std::endl;
20     }
21 };
22
23 int main() {
24     Sınıfım nesne;
25     nesne.publicVar = 5; // Genel erişimli değişkene erişim
26     nesne.publicFunc(); // Genel erişimli fonksiyonu çağırma
27
28     // privateVar ve privateFunc sınıfın dışından erişilemez:
29     // obj.privateVar = 10; // Hata! Özel erişimli değişkene erişim
30     // obj.privateFunc(); // Hata! Özel erişimli fonksiyonu çağırma
31
32     // protectedVar ve protectedFunc sınıfın dışından erişilemez:
33     // obj.protectedVar = 15; // Hata! Korunmalı erişimli değişkene erişim
34     // obj.protectedFunc(); // Hata! Korunmalı erişimli fonksiyonu çağırma
35
36     return 0; }
```

Resim 11. C++ Kapsülleme Kod Örneği

### 4.2.2 Kalıtım (Inheritance)

Kalıtım konusu nesneye yönelik programlamanın en önemli kavramlarından bir tanesidir. Kalıtım kavramı, bir sınıftan diğer bir sınıfın türetilmesidir. Yeni türeyen alt sınıf, üst sınıfın global alanlarına ve metotlarına otomatik olarak sahip olur (private olanlar hariç).

**4.2.2.1. Tekli Kalıtım :** Tekli mirasta, bir sınıfın yalnızca bir sınıftan miras almasına izin verilir. Yani bir alt sınıf yalnızca bir temel sınıf tarafından miras alınır [7].



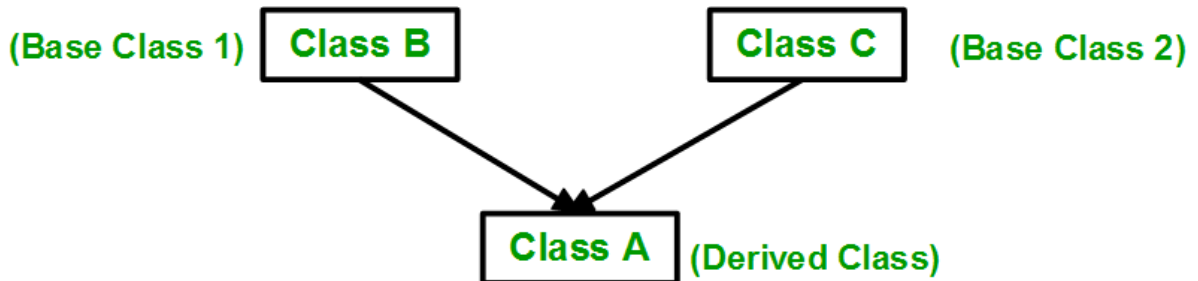
Resim 11. C++ Tekli Kalıtım

### Tekli Kalıtım Kod Örneği:

```
1  #include <iostream>
2
3
4  class Hayvan { // Ana sınıf
5  public:
6      void sesCikar() {
7          std::cout << "Hayvan: ses çıkardı!" << std::endl;
8      }
9  };
10
11
12  class Kopek : public Hayvan {
13  // Tekli kalıtım ile türetilmiş sınıf
14  public:
15      void havla() {
16          std::cout << "Köpek: Hav hav!" << std::endl;
17      }
18  };
19
20  int main() {
21      Kopek k;
22      k.sesCikar(); // Hayvan sınıfından kalıtım aldığı metod
23      k.havla();    // Kendi metodunu kullanma
24
25      return 0;
26  }
27
```

Resim 12. C++ Tekli Kalıtım Kod Örneği

**4.2.2.2 Çoklu Kalıtım:** Çoklu Kalıtım, bir sınıfın birden fazla sınıftan miras alabildiği C++'ın bir özelliğidir. yani bir alt sınıf birden fazla temel sınıftan miras alınır [7].



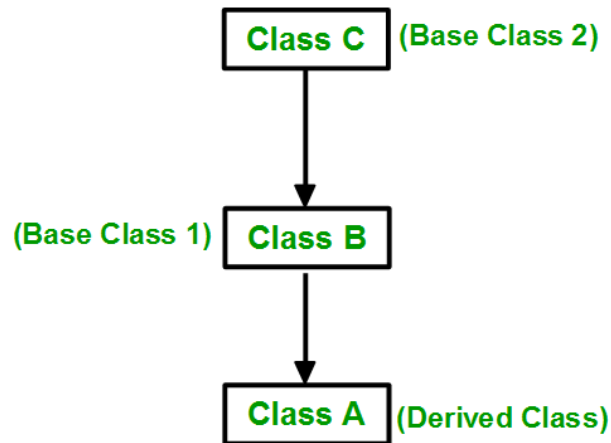
Resim 13. C++ Çoklu Kalıtım

### Çoklu Kalıtım Kod Örneği:

```
1  #include <iostream>
2
3  class Hayvan {
4  public:
5      void sesCikar() {
6          std::cout << "Hayvan: ses çıkardı!" << std::endl;
7      }
8  };
9
10 class DortAyaklıHayvan {
11 public:
12     void yuru() {
13         std::cout << "Kuş: Uçtu!" << std::endl;
14     }
15 };
16
17 class Kopek : public Hayvan, public DortAyaklıHayvan {
18 public:
19     void havla() {
20         std::cout << "Köpek: Hav hav!" << std::endl;
21     }
22 };
23
24 int main() {
25     Kopek k;
26     k.sesCikar(); // Hayvan sınıfından kalıtım aldığı metod
27     k.yuru();     // Kus sınıfından kalıtım aldığı metod
28     k.havla();    // Kendi metodunu kullanma
29
30     return 0;}
```

Resim 14. C++ Çoklu Kalıtım Kod Örneği

**4.2.2.3. Çok Düzeyli Kalıtım:** Bu miras türünde, başka bir türetilmiş sınıftan türetilmiş bir sınıf oluşturulur [7].



Resim 15. C++ Çoklu Düzeyli Kalıtım

### Çok Düzeyli Kalıtım Kod Örneği:

```
1  #include <iostream>
2
3  class Hayvan {
4  public:
5      void sesCikar() {
6          std::cout << "Hayvan: ses çıkardı!" << std::endl;
7      }
8  };
9
10 class Kus : public Hayvan {
11 public:
12     void uc() {
13         std::cout << "Kuş: Uçtu!" << std::endl;
14     }
15 };
16
17 class Kartal : public Kus {
18 public:
19     void kartalSesi() {
20         std::cout << "Kartal: Kartal Ses Çıkardı!" << std::endl;
21     }
22 };
23
24 int main() {
25     Kartal k;
26     k.sesCikar(); // Hayvan sınıfından kalıtım aldığı metod
27     k.uc();       // Kus sınıfından kalıtım aldığı metod
28     k.kartalSesi(); // Kendi metodunu kullanma
29
30     return 0;
31 }
```

Resim 16. C++ Çoklu Düzeyli Kalıtım Kod Örneği

### 4.2.3 Polimorfizm (Polymorphism)

Polimorfizm, nesneye yönelik programlamanın önemli kavramlarından biridir ve sözlük anlamı olarak “bir çok şekil”, “çok biçimlilik” anlamına gelmektedir. Aynı temel sınıftan türetilmiş olan metodun farklı sınıflarda farklı şekillerde uyarlanabilmesidir. Polimorfizm ve kalıtım konusu iç içedir.

#### Polimorfizm kod örneği:

```
1  #include <iostream>
2
3  class Hayvan { // Ana sınıf
4  public:
5      virtual void sesCikar() {
6          std::cout << "Hayvan ses çıkardı" << std::endl;
7      }
8  };
9
10 class Kedi : public Hayvan { // Türetilmiş sınıf
11 public:
12     void sesCikar() override {
13         std::cout << "Kedi miyavladı" << std::endl;
14     }
15 };
16
17 class Kopek : public Hayvan { // Türetilmiş sınıf
18 public:
19     void sesCikar() override {
20         std::cout << "Köpek Havladı" << std::endl;
21     }
22 };
23
24 int main() {
25     Hayvan* hayvan1 = new Kedi();
26     Hayvan* hayvan2 = new Kopek();
27     hayvan1->sesCikar(); // Kedi sınıfının ses çıkarma metodu
28 }
```

Resim 17. C++ Polimorfizm Kod Örneği



#### 4.2.4 Soyutlama (Abstraction)

Soyut sınıfın metotları, alt sınıfları tarafından üzerine yazılmak için (override etmek için) oluşturulan şablonlar olarak tanımlanabilir. Soyut metotların gövdesi bulunmaz. İçi boş hiçbir iş yapmayan metot görünümündedir. Bu sınıflar nesnenin “Ne?” yapmasını söyler, “Nasıl?” yapmasını gerektiği söylemez. Soyut bir sınıftan türetilmiş alt sınıflar, bu soyut sınıfın içerisindeki soyut metotları kesin olarak iptal etmeleri (override) gerekmektedir.

##### Soyutlama Kod Örneği:

```
1  #include <iostream>
2
3  class Sekil { // Soyut sınıf (abstract class)
4  public:
5      virtual void alanHesapla() = 0;
6      virtual void cevreHesapla() = 0;
7  };
8
9  class Dikdortgen : public Sekil {
10 private:
11     int uzunluk, genislik;
12
13 public:
14     Dikdortgen(int u, int g) : uzunluk(u), genislik(g) {}
15
16     void alanHesapla() override {
17         std::cout << "Dikdörtgen: Alan = " << uzunluk * genislik << std::endl;
18     }
19
20     void cevreHesapla() override {
21         std::cout << "Dikdörtgen: Çevre = " << 2 * (uzunluk + genislik) << std::endl;
22     }
23 };
24
25 int main() {
26     Dikdortgen dikdortgen(4, 5);
27     dikdortgen.alanHesapla();
28     dikdortgen.cevreHesapla();
29
30     return 0;
31 }
```

Resim 18. C++ Soyutlama Kod Örneği

## 5. Platform ve Kodlama

C++ , genel amaçlı bir programlama ve kodlama dilidir. C++, tarayıcılar, işletim sistemleri, uygulamalar geliştirme, oyun programlaması, yazılım mühendisliği, veri yapıları vb. alanlarda kullanılmaktadır.

### 5.1 C++ Kullanım Alanları

#### 5.1.1. İşletim Sistemleri

Genel olarak işletim sistemlerinin sistem kaynaklarını yönetmede hızlı ve verimli olması gerekir. C++'ın çok çeşitli sistem düzeyinde işlevlere sahip yüksek hızlı bir programlama dili olması, donanım kaynaklarına erişim ve doğrudan bellek yönetimi yeteneği, modülerlik ve nesne yönelimli programlamaya uygunluk, geniş kütüphane desteğine sahip olması işletim sistemleri geliştirmek için oldukça uygun bir tercih haline getirir. Bu durum, işletim sistemlerinin karmaşık yapısını oluştururken gerekli olan güçlü ve optimize edilmiş bir çerçeve sunar. Microsoft Windows, Apple Mac OS X gibi çoğu İşletim Sistemi ve hatta IOS gibi mobil işletim sistemleri C++ kullanılarak geliştirilmiştir [10].

#### 5.1.2. Oyunlar

Düşük-orta seviyeli bir dil olması sebebiyle C++ oyun geliştirmede en yaygın kullanılan programlama dillerinden biridir. Grafiksel oyunlar, C++'ın yüksek hızından yararlanır. Oyun oynamak, çok oyunculu oturumları kolaylaştırmak için çok sayıda kaynak yoğun işlev gerektirir. C++ kaynakları en iyi şekilde kullanabilir, çok oyunculu ağ iletişimi sağlayabilir ve 3D oyunların karmaşıklıklarını giderir. C++; World of Warcraft, Counter-Strike, Assassin's Creed serileri, Civilization IV, Half-Life 2 gibi popüler oyunların, Unreal Engine gibi oyun motorlarının ve Xbox, PlayStation ve Nintendo Switch gibi oyun konsollarının oluşturulmasında kullanılmaktadır [9].

#### 5.1.3. GUI Tabanlı Uygulamalar

C++'ın yüksek hızı, birçok GUI tabanlı ve masaüstü uygulamalarının tercih ettiği bir neden olmuştur. Örneğin, Adobe'nin Illustrator, Photoshop ve Winamp Media Player gibi Adobe sistemlerinin birçoğu C++ kullanılarak geliştirilmiştir [10].

#### 5.1.4. Tarayıcılar

C++, yüksek performansı ve gecikmeyi en aza indirme yeteneğiyle bilinen bir dil olduğundan, tarayıcı geliştirilmesinde tercih edilen bir dil olmuştur. Aynı zamanda tarayıcıların efektif bir biçimde kullanılabilmesi için cross platform yapıya sahip olmalıdır. C++, sunmuş olduğu gelişmiş donanım optimizasyonu sayesinde tarayıcıların farklı donanım ve işletim sistemleri üzerinde çalışabilmeyi sağlar. Google Chrome, Firefox, Mozilla ve Opera gibi popüler tarayıcılar C++ kullanılarak geliştirilmiştir [9].

### 5.1.5. Veritabanı Yönetim sistemi (Database Management System)

C++'ın dosya işleme, yüksek hız, güvenilirlik, işlevler, sınıflar ve nesneler gibi özellikleri onu veri yönetimi için ideal kılar. MySQL, Oracle, Postgres ve MongoDB gibi popüler veritabanı yönetim araçları C++ ile geliştirilmiştir [10].

### 5.1.6. Bulut/Dağıtık Sistemler (Cloud Computing and Distributed Applications)

Düşük-orta seviyeli bir dil olması sebebiyle C++, bulut depolama sistemlerini uygulamak için tercih edilen bir dil haline gelir. Serverlarla oldukça uyumlu olması, hızlı bir dil olması, bellek yönetiminin güçlü olması nedeniyle bulut bilişimi ve dağıtılmış uygulamaları kullanan büyük şirketler, donanım sistemine yük toleransının yanı sıra eş zamanlı uygulamalar oluşturabilen çoklu iş parçacığı desteği sağladığı için C++ kullanıyorlardır [10]. Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP) gibi platformların geliştirilmesinde C++ kullanılmıştır [9].

### 5.1.7. Libraries

Kütüphaneler çok üst düzey matematiksel hesaplamalar, performans ve hız gerektirir. C++; standart kütüphanesi çok çeşitli yerleşik fonksiyonlara sahip olması, yüksek hızlı, performans düzeyi yüksek olan bir dil olmasından dolayı tercih kütüphane geliştirilmesinde tercih edilen bir dil olmuştur. En popüler Makine Öğrenimi kütüphanelerinden biri olan Tensorflow, arka uç (backend) programlama dili olarak C++ kullanır.

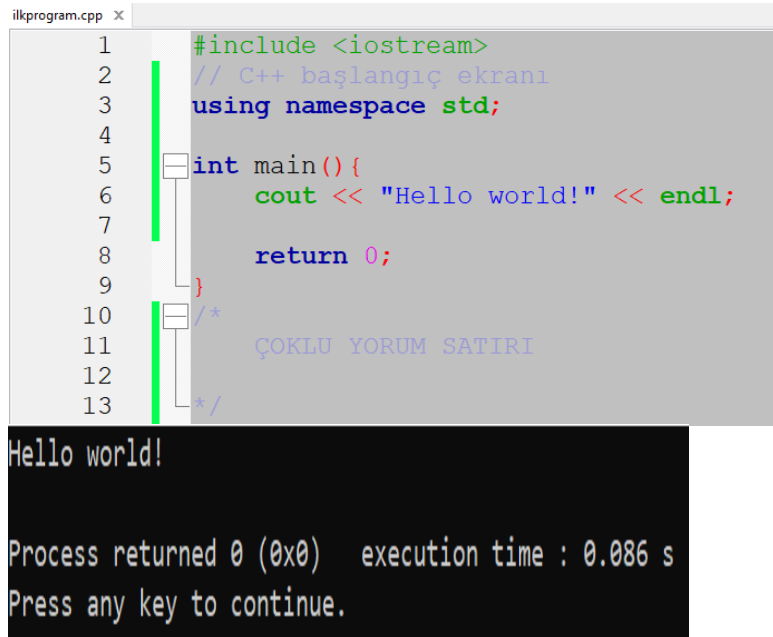
### 5.1.8. Gömülü Sistemler

C++, akıllı saatlerden tıbbi ekipmanlara kadar çeşitli gömülü sistemlerin geliştirilmesinde tercih edilir [10]. Diğer üst düzey dillere kıyasla, C++ donanım özelliklerine doğrudan erişim sağlayarak düşük seviyeli işlevleri kolayca kullanabilir. C++'ın bir alt kümesi olan Gömülü C++ (Embedded C++), gömülü sistem programlama açısından kod boyutunu en aza indirmeyi ve yürütme verimliliğini en üst düzeye çıkarmayı amaçlamaktadır [11].

### 5.1.9. Derleyiciler

Derleyiciler, bellek yönetimi, hız ve hata tespiti açısından güçlü bir yapıya sahip olmalıdır. C++, düşük ve orta seviyeli bir dil olarak yüksek performans, hızlı çalışma ve güçlü optimizasyon yetenekleri sunması nedeniyle derleyici geliştirmekte tercih edilen bir dil haline gelmiştir. Ayrıca, derleyicilerin düşük seviyeli bir dille yazılmasının işlemcilerle sunduğu bazı avantajlar bulunur. Örneğin, derleyicilerin düşük seviyeli dilde yazılması, işlemcilerle daha yakın bir ilişki kurabilmelerini sağlar. Böylece, donanımın potansiyelini daha iyi değerlendirebilirler ve performansı artırabilirler. GCC(GNU Compiler Collection), Intel C++ Compiler, Microsoft Visual C++ Compiler C++ kullanarak geliştirilmiştir [9].

## 5.2 C++'da Kodlama



```
1 #include <iostream>
2 // C++ başlangıç ekranı
3 using namespace std;
4
5 int main() {
6     cout << "Hello world!" << endl;
7
8     return 0;
9 }
10 /*
11     ÇOKLU YORUM SATIRI
12 */
13
```

Hello world!

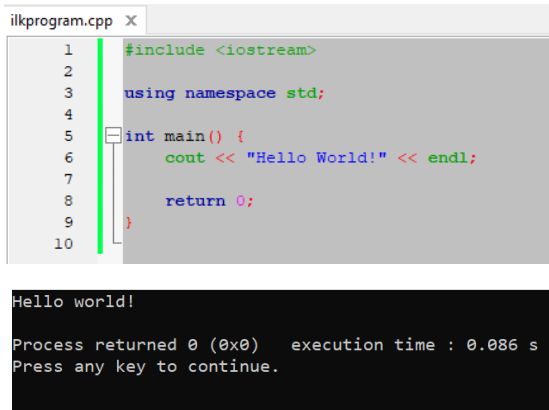
Process returned 0 (0x0) execution time : 0.086 s  
Press any key to continue.

Resim 19. C++ Dosyası Karşılama Kodu

**Satır 1:** `#include <iostream>` giriş/çıkış (I/O) işlemleri yapmak için kullanılan bir kitaplıktır.

**Satır 2:** Tek satırlık yorumlar iki eğik çizgiyle ( `//` ) başlar. Yorum satırları derleyici tarafından dikkate alınmaz (yürütülmez).

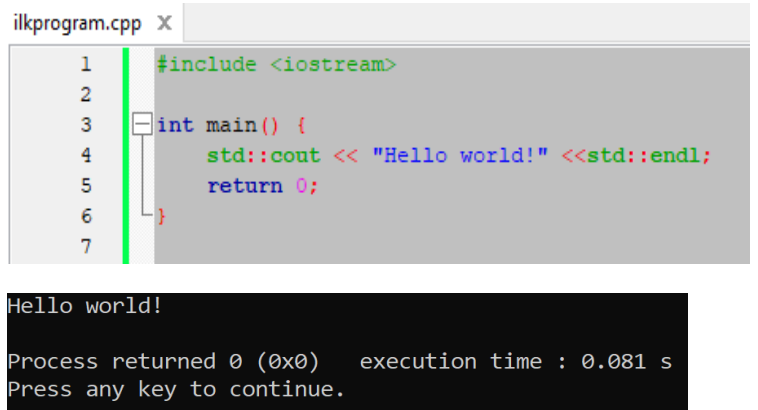
**Satır 3:** `using namespace std;` standart kütüphanedeki nesneler ve değişkenler için adları kullanabileceğimiz anlamına gelir. Bu satır yazıldığında kitaplıklar içindeki işlevleri (fonksiyonlar, sınıflar vb. ) kullanırken her seferinde `std::` kullanılmasına gerek yoktur.



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello World!" << endl;
7
8     return 0;
9 }
10
```

Hello world!

Process returned 0 (0x0) execution time : 0.086 s  
Press any key to continue.



```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello world!" <<std::endl;
5     return 0;
6 }
7
```

Hello world!

Process returned 0 (0x0) execution time : 0.081 s  
Press any key to continue.

Resim 20. Resim 21. C++ namespace kullanımı

**Satır 5:** Bir C++ kodu çalışmaya `main()` fonksiyonundan başlar. Bu fonksiyon olmadan bir C++ programı geçerli bir C++ programı olamaz. Süslü parantez ( `{ }` ) içerisinde yazılı olan kod bloğu yürütülecektir. C++ standardına göre `main()` fonksiyonu kesinlikle `int` türünde bir değer döndürmelidir.

**Satır 6:** C++’da süslü parantezler ({} ) blokları belirtmek için kullanılır. Bir blok bir veya daha fazla ifadeyi barındırır ve bu ifadeleri birbirinden ayırarak okunabilirliği artırır.

**Satır 7:** *cout* iostream kitaplığı içerisinde bulunan nesnedir. Ekrana çıktı vermek için kullanılan bir standart çıkış akış nesnesidir. Akış yönlendirmesi operatörü (<<) , sağdaki değeri soldaki nesneye yönlendirir. Karakter dizileri tırnak içinde ("" ) yazılır. *endl* standart çıkış akış nesnesi, yeni satır eklemek için kullanılır. *endl* nesnesine alternatif olarak (\n) ifadesi de kullanılabilir.

Örneğin; `cout << "Hello World! \n";`

**Satır 8:** *return* ifadesi, bir C++ fonksiyonundan değer döndürmek için kullanılır. Fonksiyonun geri döndürdüğü değer, fonksiyonun başındaki belirtilen türle uyumlu olmalıdır.

**Satır 10-13:** Çok satırlı yorumlar /\* ile başlar ve \*/ ile biter .

## 5.3 C++’da Operatörler

C++ programlama dilinde operatörler, değişkenler ve değerler arasında belirli işlemleri gerçekleştirmek için kullanılan sembollerdir. Bu operatörler aritmetik, karşılaştırma, mantıksal atama ve bitisel gibi farklı kategorilere ayrılır. Aritmetik operatörler matematiksel hesaplamalar için kullanılırken, karşılaştırma operatörler değerler arasındaki ilişkileri kontrol eder. Mantıksal operatörler ise mantıksal ifadeleri değerlendirir ve atama operatörleri bir değeri bir değişkene atamak için kullanılır. Operatörler, programların işleyişini kontrol etmeye yardımcı olur.

### 5.3.1 Aritmetik Operatörler

Aritmetik operatörler genel matematiksel işlemleri gerçekleştirmek için kullanılır.

Operator	Anlam	Tanım	Örnek
+	Addition	İki değişkeni toplar	x + y
-	Subtraction	Bir değeri diğerinden çıkarır	x - y
*	Multiplication	İki değeri çarpar	x * y
/	Division	Bir değeri diğerine böler	x / y
%	Modulus	Mod alır	x % y
++	Increment	Bulunduğu konuma göre değişkenin değerini atamadan önce veya sonra bir artırır.	++x
--	Decrement	Bulunduğu konuma göre değişkenin değerini atamadan önce veya sonra bir azaltır.	--x

**Resim 22. C++ Aritmetik Operatörler**

### 5.3.2 Karşılaştırma Operatörleri

Karşılaştırma operatörleri iki değeri (veya değişkeni) karşılaştırmak için kullanılır. Bu operatörler, iki değeri karşılaştırır ve karşılaştırmanın sonucunu bir mantıksal (boolean) değer olan true veya false olarak döndürür.

Operator	Anlam	Örnek
==	Eşittir	x == y
!=	Eşit değildir	x != y
>	Büyüktür	x > y
<	Küçüktür	x < y
>=	Büyük eşittir	x >= y
<=	Küçük eşittir	x <= y

Resim 23. C++ Karşılaştırma Operatörleri

### 5.3.3 Mantıksal Operatörler

Mantıksal ifadeleri değerlendirmek ve kontrol etmek için kullanılan sembollerdir. Karşılaştırma operatörlerinde olduğu gibi mantıksal operatörler de true veya false değerleri döndürür.

Operator	Anlam	Tanım
&&	Mantıksal ve	Her iki ifade de doğruysa true değeri döndürür
	Mantıksal veya	İfadelerden biri doğruysa true değeri döndürür.
!	Mantıksal değil	Sonucu tersine çevirir, sonuç doğruysa false değeri döndürür.

Resim 24. C++ Mantıksal Operatörler

### 5.3.4 Atama Operatörleri

Değişkenlere değer atamak için kullanılan operatörlerdir. Bir değişkene değer atamak için kullanılır. Atama işlemleri her zaman sağdan sola olacak şekilde gerçekleşir.

Operator	Örnek	Açılım
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

Resim 25. C++ Atama Operatörleri

### 5.3.5 Bitisel Operatörler

Bitisel operatörler, tam sayıların bitlerini doğrudan manipüle etmek için kullanılan operatörlerdir.

x	y	x y	x&y	x^y
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Resim 26. C++ Bitisel Operatörler

## 5.4 C++ Veri Tipleri

C++'da veri tipleri, değişkenlere atanabilecek verilerin türlerini belirleyen yapıları ifade eder. Bu veri tipleri, tam sayıları, ondalık sayıları, karakterleri, metinleri ve diğer değerleri temsil eder. Her veri tipi, belirli bir bellek miktarını tutar. Veri tipleri, değişkenin boyutunu, aralığını ve nasıl kullanılabileceğini belirler.

Data Type	Meaning	Size (in Bytes)
int	Integer	2 or 4
float	Floating-point	4
double	Double Floating-point	8
char	Character	1
bool	Boolean	1
void	Empty	0

**Resim 27. C++ Veri Tipleri**

## Örnek Kullanım:

```

/*
Sözdizimi=> tip değişkenAdı=değer
*/

int sayi = 5;           // Integer (tam sayı)
float floatSayi = 2.25; // Kayan Nokta Sayılar
double doubleSayi = 5.99; // Kayan Nokta Sayılar
char harf = 'D';        // Karakter
bool durum = true;      // True veya False
void durum;             // Bos
string metin = "Hello"; // String (metin)

```

**Resim 28. C++ Veri Tipleri Kullanımı**

## 5.5 C++'da Kaçış Dizileri

C++'da kaçış operatörleri, karakter dizilerine özel karakterler eklemek için kullanılır. Kaçış operatörleri, karakter dizilerinin içeriğini biçimlendirmek ve standart olmayan karakterleri temsil etmek için kullanılır. Escape (\) karakteri özel karakterleri yazdırmak için kullanılır.

**Tablo 1. C++ Kaçış Dizileri**

'\n'	Yeni satır karakteridir. Ekranda bir alt satıra geçmeyi sağlar.
'\\'	Metin içinde ters eğik çizgiyi göstermek için kullanılır.
'\b'	Geri tuşu (backspace) karakteridir. Bir karakteri geri almayı sağlar.
'\t'	Ekranda bir sekme boşluğu bırakmayı sağlar.
'\r'	Carriage return karakteridir. İlgili satırın başına dönmeyi sağlar. Genellikle dosya işlemlerinde kullanılır.
'/*'	Metin içinde çift tırnak karakterini göstermek için kullanılır.
'/a'	Uyarı sesi çıkarmak için kullanılır.



## 5.6 C++ hangi platformlarda yazılabilir?

C++ programlama dilinin taşınabilir özelliği, çeşitli platformlarda çalışabilme yeteneğine olanak tanır. Bu özellik, Windows, MacOS, Linux gibi farklı işletim sistemlerinde sorunsuz bir şekilde kullanılabilmesine olanak sağlar. C++ dilindeki bu taşınabilirlik, yazılım geliştiricilere uygulamalarını farklı ortamlarda rahatlıkla dağıtabilme esnekliği sunar. C++ dilindeki kodları geliştirmek için farklı entegre geliştirme ortamları (IDE) kullanılabilir. Bunlar arasında Microsoft Visual Studio, Code::Blocks, CLion, C++ gibi IDE'ler bulunmaktadır.

# Geliştirilen Proje

Github linki : <https://github.com/SemaEkmekci/Qt-Creator--H-Langman>

## langman.h

```
1  #ifndef LANGMAN_H
2  #define LANGMAN_H
3
4  #include <QMainWindow>
5  #include <QTimer>
6  #include <QtMultimedia>
7  #include <QAudioOutput>
8  #include <QPushButton>
9  #include "word.h"
10
11
12  QT_BEGIN_NAMESPACE
13  namespace Ui { class Langman; }
14  QT_END_NAMESPACE
15
16  class Langman : public QMainWindow
17  {
18  |   Q_OBJECT
19
20  public:
21  |   Langman(QWidget *parent = nullptr);
22  |   ~Langman();
23
24
25  private slots:
26  |   void backgroundSound();
27  |   void startGameBtn_Clicked();
28  |   void startGameBtn_2_Clicked();
29  |   void howToPlayBtn_Clicked();
30  |   void nextLevelBtnClicked();
31  |   void buttonHover();
32  |   void volumeBtn();
33  |   void sarkiBitti();
34  |   void backBtn();
35  |   void randomUserName();
36  |   void rightGuessVoice();
37  |   void wrongGuessVoice();
38  |   void toWinVoice();
39  |   void toLoseVoice();
40  |   void trueWordVoice();
41  |   void trueWord();
42  |   void keyPress();
43  |   void keyPressEvent(QKeyEvent *event);
44  |   void gameScreen();
45  |   void getLetter();
46  |   void toWin();
47  |   void showClue();
48  |   void showClueLetter();
49  |   void showInfo();
50  |   void manDrawNotVisible();
51  |   void manDraw(int count);
52  |   void gameReset();
53  |   void newGameBtn();
54
55
56  private:
57  |   Ui::Langman *ui;
58  |   QMediaPlayer *player; //Arka plan sesi için
59  |   QMediaPlayer *player2; //Buton sesleri için
60  |   QMediaPlayer *player3; //Klavye sesleri için
61  |   QMediaPlayer *player4; //Game Over için
62  |   QAudioOutput *audioOutput;
63  |   QAudioOutput *audioOutput_2; //Arka plans sesi hariç diğer seslerin düzeyini ayarlamak için
64  |   Letter *let; // Letter sınıfından nesne türetiyorum.
65  |   QTimer *timer;
66  |   QLabel *letter;
67  |   QLabel *line;
68  |   QStringList falseLetter; //Yanlış harfleri ekranda göstermek için
69  |   QString falseLets = "";
70  |   QString trueLets = "";
71  |   int trueCount; //Kelime bilindi mi?
72  |   int length; //Kelimenin uzunluğu
73  |   bool isSoundOn = true;
74  |   int clueCount = 2;
75  |   int score = 0;
76  |   int falseCount = 0; //Adam çizmek için yanlış harf sayacı
77  };
78  #endif // LANGMAN_H
79
```

## langman.cpp

```
1 #include "langman.h"
2 #include "ui_langman.h"
3 #include <QMessageBox>
4 #include <fstream>
5 #include <QBoxLayout>
6 #include <QLabel>
7 #include <QStringList>
8
9 QVector<QLabel> letterLabels;
10 QVector<QLabel> lineLabels;
11 QStringList questionInfo;
12
13 Langman::Langman(QWidget *parent)
14 : QMainWindow(parent)
15 , ui(new Ui::Langman)
16 {
17     setFixedSize(1021, 681);
18     ui->setUpUi(this);
19     manDrawMotVisible();
20     ui->stackedWidget->setCurrentIndex(0); //Anasayfa
21     backgroundSound();
22     timer = new QTimer(this); //Arka planda çalan şarkı bitince tekrar dönsün diye oluşturdum.
23     timer->start(110000);
24
25     isSoundOn = true;
26
27
28     connect(ui->startGame, &QPushButton::clicked, this, &Langman::startGameBtn_Clicked);
29     connect(ui->backHowBtn, &QPushButton::clicked, this, &Langman::newGameBtn);
30     connect(ui->backGameBtn, &QPushButton::clicked, this, &Langman::newGameBtn);
31     connect(ui->startNewGame, &QPushButton::clicked, this, &Langman::newGameBtn);
32     connect(ui->nextLevel, &QPushButton::clicked, this, &Langman::nextLevelBtn_Clicked);
33     connect(ui->startGame_2, &QPushButton::clicked, this, &Langman::startGameBtn_2_Clicked);
34     connect(ui->sugges, &QPushButton::clicked, this, &Langman::randomUserName);
35     connect(ui->volumeBtn, &QPushButton::clicked, this, &Langman::volumeBtn);
36     connect(ui->startGame, &QPushButton::clicked, this, &Langman::buttonHover);
37     connect(ui->howToPlay, &QPushButton::clicked, this, &Langman::buttonHover);
38     connect(ui->backHowBtn, &QPushButton::clicked, this, &Langman::buttonHover);
39     connect(ui->backGameBtn, &QPushButton::clicked, this, &Langman::buttonHover);
40     connect(ui->startNewGame, &QPushButton::clicked, this, &Langman::buttonHover);
41     connect(ui->volumeBtn, &QPushButton::clicked, this, &Langman::buttonHover);
42     connect(ui->clue, &QPushButton::clicked, this, &Langman::buttonHover);
43     connect(ui->clue, &QPushButton::clicked, this, &Langman::showClueLetter);
44
45     connect(ui->backBtn, &QPushButton::clicked, this, &Langman::buttonHover);
46     connect(ui->backBtn, &QPushButton::clicked, this, &Langman::backBtn);
47     //connect(ui->backBtn, &QPushButton::clicked, this, &Langman::buttonHover);
48
49     connect(ui->startGame_2, &QPushButton::clicked, this, &Langman::buttonHover);
50     connect(ui->name, &QLineEdit::textChanged, this, &Langman::keyPress); //Oyuna başla sayfasındaki isim alanına isim girilirken klavye sesi çıkması için
51     connect(ui->sugges, &QPushButton::clicked, this, &Langman::buttonHover);
52     connect(timer, SIGNAL(timeout()),this,SLOT(sarkıBitti()));
53
54 }
55
56 ~Langman::~Langman()
57 {
58     delete ui;
59 }
60
61 void Langman::backgroundSound()
62 {
63     player = new QMediaPlayer;
64     audioOutput = new QAudioOutput;
65     player->setAudioOutput(audioOutput);
66     audioOutput->setVolume(0.3);
67     player->setSource(QUrl("qrc:/new/media/spook.mp3"));
68     player->play();
69     audioOutput_2 = new QAudioOutput;
70     audioOutput_2->setVolume(1);
71 }
72
73 void Langman::startGameBtn_Clicked()
74 {
75     ui->warning->setText("");
76     ui->name->setText("Adınızı Giriniz");
77     ui->stackedWidget->setCurrentIndex(1); //Oyuna Giriş Ekranı
78     ui->win->setVisible(false);
79 }
80
81 void Langman::startGameBtn_2_Clicked()
82 {
83     qDebug() << "text:" << ui->name->text();
84     if(ui->name->text() == "" || ui->name->text() == "Adınızı Giriniz") //isim kontrolü
85     {
86         ui->warning->setText("Bu İsim Geçerli Değil !");
87     }
88     else if(ui->name->text().length() > 20)
89     {
90         ui->warning->setText("İsim 20 karakterden fazla olamaz !");
91     }
92
93     else
94     {
95         qDebug() << "text:" << ui->name->text();
96         ui->nameLabel->setText("Oyuncu: " + ui->name->text());
97         ui->clueInfo->setText("");
98         ui->score->setText("Skor: 0");
99         score = 0;
100         if(clueCount != 0){
101             ui->clue->setDisabled(false);
102         }
103         else{
104             ui->clue->setDisabled(true);
105         }
106
107         ui->stackedWidget->setCurrentIndex(2); //Oyun Ekranı
108         gameScreen(); //Letter sınıfına gidip json dosyasından veri alıyorum.
109         getLetter();
110     }
111 }
```

```

111 }
112 }
113
114 void Langman::howToPlayBtn_Clicked()
115 {
116     ui->stackedWidget->setCurrentIndex(5); //Nasıl Oynanır Ekranı
117 }
118
119 void Langman::newGameBtn()
120 {
121     ui->stackedWidget->setCurrentIndex(0); //Anasayfa
122 }
123
124 void Langman::nextLevelBtn_Clicked()
125 {
126     ui->stackedWidget->setCurrentIndex(2);
127     getLetter();
128 }
129
130 void Langman::buttonHover()
131 {
132     player2 = new QMediaPlayer;
133     player2->setAudioOutput(audioOutput_2);
134     player2->setSource(QUrl("qrc:/new/media/mouse-click.mp3"));
135     player2->play();
136 }
137
138 void Langman::volumeBtn()
139 {
140     if (isSoundOn) {
141         player->stop();
142         ui->volumeBtn->setIcon(QIcon(":/new/image/volumeDisabled.png"));
143         isSoundOn = false;
144     } else {
145         player->play();
146         ui->volumeBtn->setIcon(QIcon(":/new/image/volume.png"));
147         isSoundOn = true;
148     }
149 }
150
151 void Langman::sarkiBitti()
152 {
153     if (isSoundOn) {
154         player->stop();
155         player->setPosition(0);
156         player->play();
157     }
158 }
159
160

```

```

161 void Langman::backBtn()
162 {
163     if (QMessageBox::question(this, "", "Oyundan çekilmek istiyor musunuz?", QMessageBox::Yes|QMessageBox::No) == QMessageBox::Yes)
164     {
165         ui->stackedWidget->setCurrentIndex(0);
166         gameReset();
167     }
168 }
169
170
171 void Langman::randomUserName()
172 {
173     QFile readFile("~/new/files/username.txt");
174     if(readFile.open(QIODevice::ReadOnly | QIODevice::Text)){
175         int lineNum = 1;
176         int random = rand() % 25 + 1; //1 ile 25 arasında bir sayı ürettiyorum. Çünkü isim dosyasında 25 satır var.
177         QTextStream in(&readFile);
178         QString line;
179         while(!in.atEnd()){ // Bu döngü random üretilen sayısının olduğu satıra gelince sonlanacak.
180             line = in.readLine();
181             lineNum++;
182             if(lineNum == random)
183             {
184                 break;
185             }
186         }
187         readFile.close();
188         qDebug() << "line: " << line;
189         ui->name->setText(line);
190         readFile.close();
191     }
192     else
193     {
194         qDebug() << "Üzgünüm isim yok";
195     }
196 }
197
198
199 void Langman::rightGuessVoice()
200 {
201     player3 = new QMediaPlayer;
202     player3->setAudioOutput(audioOutput_2);
203     player3->setSource(QUrl("qrc:/new/media/correct-choice.mp3"));
204     // audioOutput->setVolume(50);
205     player3->play();
206 }
207
208
209
210

```

```

210 }
211
212 void Langman::wrongGuessVoice()
213 {
214     player3 = new QMediaPlayer;
215     player3->setAudioOutput(audioOutput_2);
216     player3->setSource(QUrl("qrc:/new/media/negative_beeeps.mp3"));
217     player3->play();
218 }
219
220 void Langman::toWinVoice()
221 {
222     player3 = new QMediaPlayer;
223     player3->setAudioOutput(audioOutput_2);
224     player3->setSource(QUrl("qrc:/new/media/success-trumpets.mp3"));
225     player3->play();
226 }
227
228 void Langman::toLoseVoice()
229 {
230     player3 = new QMediaPlayer;
231     player3->setAudioOutput(audioOutput_2);
232     player3->setSource(QUrl("qrc:/new/media/wahwahsadtrumbone.mp3"));
233     player3->play();
234 }
235
236 void Langman::trueWordVoice()
237 {
238     player3 = new QMediaPlayer;
239     player3->setAudioOutput(audioOutput_2);
240     player3->setSource(QUrl("qrc:/new/media/success.mp3"));
241     player3->play();
242 }
243
244 void Langman::trueWord()
245 {
246     trueWordVoice();
247     QTimer::singleShot(500, [=]() { //Ekranda girilen son harfi de bir süre görebilmek için biraz bekletiyorum.
248         ui->stackedWidget->setCurrentIndex(3);
249         trueLets=""; //Oyunda kullanılan harfleri sıfırlıyorum. Çünkü oyun yeniden açılmış gibi oluyor.
250         ui->word->setText("");
251
252         for (QLabel* label : lineLabels) //Kelime ile işim bittiğinde listemi temizliyorum.
253         {
254             delete label;
255         }
256         for (QLabel* label : letterLabels) //Kelime ile işim bittiğinde listemi temizliyorum.
257         {
258             delete label;
259         }
260
261         lineLabels.clear();
262         falseLetter.clear();
263         letterLabels.clear();
264         showInfo();

```

```

264     showInfo();
265
266     });
267
268 }
269
270 void Langman::keyPress() //Klavye tuşları sesleri için
271 {
272     qDebug() << "Tuş: basıldı: ";
273     player3 = new QMediaPlayer;
274     player3->setAudioOutput(audioOutput_2);
275     player3->setSource(QUrl("qrc:/new/media/mech-keyboard.mp3"));
276     player3->play();
277 }
278
279 void Langman::keyPressEvent(QKeyEvent *event)
280 {
281
282     int currentPageIndex = ui->stackedWidget->currentIndex();
283
284     falseLets = "";
285
286     qDebug() << "Tuş" << currentPageIndex;
287     if (currentPageIndex == 2) //Klavyeden sadece harfler alınacak
288     {
289         if(trueCount == length)
290         {
291             //Eğer kelime bilindiyse fonksiyondan çık.
292             return;
293         }
294         if(event->text().length() == 1 && (event->text().at(0).isLetter() || event->text() == '#' ||
295         keyPress(); //Klavye'den sadece harfler, '+' ve '#' karakterini almak için
296         bool isFound = questionInfo[0].contains(event->text().toUpper());
297         qDebug() << "isFound" << isFound;
298
299         if (isFound) {
300
301             qDebug() << "Girilen Harf: " <<event->text();
302             if(!trueLets.contains(event->text().toUpper())){
303                 rightGuessVoice();
304                 score += 10;
305                 trueLets+=event->text().toUpper();
306                 ui->score->setText("Skor: " + QString::number(score));
307                 for (int i = 0; i < letterLabels.size(); i++)
308                 {
309                     if (questionInfo[0][i] == event->text().toUpper())
310                     {
311                         letterLabels[i]->setVisible(true);
312                         trueCount++;
313
314                         qDebug() <<"trueCount" <<trueCount;
315                         if(trueCount == length)
316                         {
317                             trueWord();
318                             return;
319                         }

```

```

318         }
319     }
320 }
321
322 }else if(!falseLetter.contains(event->text().toUpper())) //Kullanılan harf tekrar kullanılmaması için
323 {
324     falseCount++;
325     qDebug() << "falseCount: " << falseCount;
326     if(falseCount == 12) //Oyun bitti KAYBEDİLDİ.
327     {
328         toLoseVoice(); //Kaybedilirse bu ses çıkacak
329         ui->info_2->setText(questionInfo[2]); //Son kelimenin bilgisi ekranda belirecek.
330         ui->score_2->setText("Skor: " + QString::number(score)); // Skor bilgisi ekranda belirecek.
331         ui->stackedWidget->setCurrentIndex(4);
332         gameReset();
333         return;
334     }
335
336     wrongGuessVoice();
337     score-=2;
338     ui->score->setText("Skor: " + QString::number(score));
339     falseLetter.append(event->text().toUpper());
340     for (QString falseLet : falseLetter)
341     {
342         falseLets += " " + falseLet;
343     }
344     ui->word->setText(falseLets);
345     if(falseCount <= 12){ //Kullanıcının 12 defa hata yapma hakkı var.
346         manDraw(falseCount);
347     }
348 }
349 }
350 }
351 }
352
353
354 void Langman::gameScreen()
355 {
356     let = new Letter(this);
357     let->fileReadJson();
358 }
359
360 void Langman::getLetter()
361 {
362     qDebug() << "Çalıştı";
363     questionInfo = let->randomLang();
364     qDebug() << "Çalıştı2";
365     for (QLabel* label : letterLabels) //kelime ile işin bittiğinde listemi temizliyorum ki ekranda görünmesin ve crashed olmasın.
366     {
367         delete label;
368     }
369     letterLabels.clear();
370     if(questionInfo.size() == 1){ //Eğer questionInfo boyutu 1 ' e eşitse yarışma kazanılmıştır.
371         toWin();
372         return;
373     }

```

```

371         toWin();
372         return;
373     }
374     trueCount = 0;
375
376     length = questionInfo[0].length();
377     int geometry = 100;
378     int geometryLine = 100;
379     for (int i = 0; i < length; i++)
380     {
381         qDebug() << i;
382         //QString labelName = "letter_" + QString::number(length);
383         letter = new QLabel(this);
384         letter->setGeometry(geometry, 300, 100, 100);
385         geometry+=60;
386         letter->setText(questionInfo[0][i]);
387         letter->setStyleSheet("color:white; font-family: 'Papyrus', monospace; font-size: 40px;");
388         letter->font();
389         layout()->addWidget(letter);
390         letter->setVisible(false);
391         letterLabels.append(letter);
392
393         line = new QLabel(this);
394         line->setStyleSheet("border-image:url(:/new/image/line.png);");
395         line->setGeometry(geometryLine, 300, 50, 30);
396         geometryLine+=60;
397         lineLabels.append(line);
398         layout()->addWidget(line);
399     }
400
401     showClue();
402 }
403
404 void Langman::toWin()
405 {
406     questionInfo = let->randomLang();
407     ui->clueInfo->setText(""); //Oyun kazanıldıktan veya kaybedildikten sonra Ekranda ipucu görünmeyecek
408     length = questionInfo[0].length();
409     int geometry = 100;
410     int geometryLine = 100;
411     for (int i = 0; i < length; i++)
412     {
413         qDebug() << i;
414         letter = new QLabel(this);
415         letter->setGeometry(geometry, 300, 100, 100);
416         geometry+=60;
417         letter->setText(questionInfo[0][i]);
418         letter->setStyleSheet("color:white; font-family: 'Papyrus', monospace; font-size: 40px;");
419         letter->font();
420         layout()->addWidget(letter);
421         letterLabels.append(letter);
422
423         line = new QLabel(this);
424         line->setStyleSheet("border-image:url(:/new/image/line.png);");
425         line->setGeometry(geometryLine, 300, 50, 30);
426         geometryLine+=60;

```

```

425         line->setGeometry(geometryLine, 380, 50, 30);
426         geometryLine+=60;
427         lineLabels.append(line);
428         layout()->addWidget(line);
429     }
430     if(questionInfo[0] == "KAZANDINIIZ!!!"){
431         ui->win->setVisible(false);
432         toWinVoice(); //Oyun kazanılınca bu ses çalacak
433     }
434 }
435 }
436
437 void Langman::showClue()
438 {
439     ui->clueInfo->setText(questionInfo[1]);
440 }
441
442 void Langman::showClueLetter()
443 {
444     //ui->clue->setDisabled(true);
445     if(clueCount>0){
446         int random = rand() % letterLabels.size();
447         if(!trueLets.contains(letterLabels[random]->text().toUpper())){
448             clueCount--;
449             for(int i = 0; i< letterLabels.size(); i++){
450                 if(letterLabels[random]->text() == letterLabels[i]->text()){
451                     letterLabels[i]->setVisible(true);
452                     trueLets+=letterLabels[random]->text().toUpper();
453                     trueCount++;
454
455                     qDebug() <<"trueCount" <<trueCount;
456                     qDebug() <<"length" <<length;
457                     if(trueCount == length)
458                     {
459                         trueWord();
460                         return;
461                     }
462                     qDebug() << "İpucu:" << clueCount; } }
463
464     }else{
465         showClueLetter();
466     }
467 }
468 }

```

```

469 void Langman::showInfo()
470 {
471     ui->info->setText(questionInfo[2]);
472 }
473
474 void Langman::manDrawNotVisible()
475 {
476     falseCount = 0;
477     ui->man_1->setVisible(false);
478     ui->man_2->setVisible(false);
479     ui->man_3->setVisible(false);
480     ui->man_4->setVisible(false);
481     ui->man_5->setVisible(false);
482     ui->man_6->setVisible(false);
483     ui->man_7->setVisible(false);
484     ui->man_8->setVisible(false);
485     ui->man_9->setVisible(false);
486     ui->man_10->setVisible(false);
487     ui->man_11->setVisible(false);
488     ui->man_12->setVisible(false);
489     ui->eyes->setVisible(false);
490     ui->mouth->setVisible(false);
491 }
492
493 void Langman::manDraw(int count)
494 {
495     if(count == 1)
496     {
497         ui->man_1->setVisible(true);
498     }
499     else if(count == 2)
500     {
501         ui->man_2->setVisible(true);
502     }
503     else if(count == 3)
504     {
505         ui->man_3->setVisible(true);
506     }
507     else if(count == 4)
508     {
509         ui->man_4->setVisible(true);
510     }
511     else if(count == 5)
512     {
513         ui->man_5->setVisible(true);
514     }
515     else if(count == 6)
516     {
517         ui->man_6->setVisible(true);
518     }
519     else if(count == 7)
520     {
521         ui->man_7->setVisible(true);
522         ui->eyes->setVisible(true);
523     }

```



```

520     else if(count == 7)
521     {
522         ui->man_7->setVisible(true);
523         ui->eyes->setVisible(true);
524         ui->mouth->setVisible(true);
525     }
526     else if(count == 8)
527     {
528         ui->man_8->setVisible(true);
529     }
530     else if(count == 9)
531     {
532         ui->man_9->setVisible(true);
533     }
534     else if(count == 10)
535     {
536         ui->man_10->setVisible(true);
537     }
538     else if(count == 11)
539     {
540         ui->man_11->setVisible(true);
541     }
542     else if(count == 12)
543     {
544         ui->man_12->setVisible(true);
545     }
546 }
547 }
548
549 void Langman::gameReset()
550 {
551     clueCount = 2;
552     for (QLabel* label : letterLabels) //Kelime ile işim bittiğinde listemi temizliyorum.
553     {
554         delete label;
555     }
556     letterLabels.clear();
557     for (QLabel* label : lineLabels) //Kelime ile işim bittiğinde listemi temizliyorum.
558     {
559         delete label;
560     }
561     lineLabels.clear();
562     falseLetter.clear();
563     trueLet="" ; //Oyunda kullanılan harfleri sıfırlıyorum. Çünkü oyun yeniden açılmış gibi oluyor.
564     ui->word->setText("");
565     falseCount = 0;
566     manDrawNotVisible(); // Adamı ekranda görünmez hale getiriyorum.
567 }
568 }

```

## word.h

```

< > 🔒 📄 word.h
1  #ifndef WORD_H
2  #define WORD_H
3
4  #include <QWidget>
5  #include <QLabel>
6  #include <QJsonObject>
7
8  class Letter : public QLabel
9  {
10 |     Q_OBJECT
11 public:
12     explicit Letter(QWidget *parent = 0);
13     bool isItTrue;
14 public slots:
15     void fileReadJson();
16
17     QStringList randomLang();
18 private:
19     QJsonObject jsonObject;
20     QString randomKey;
21     QStringList keys;
22 };
23
24 #endif // WORD_H
25

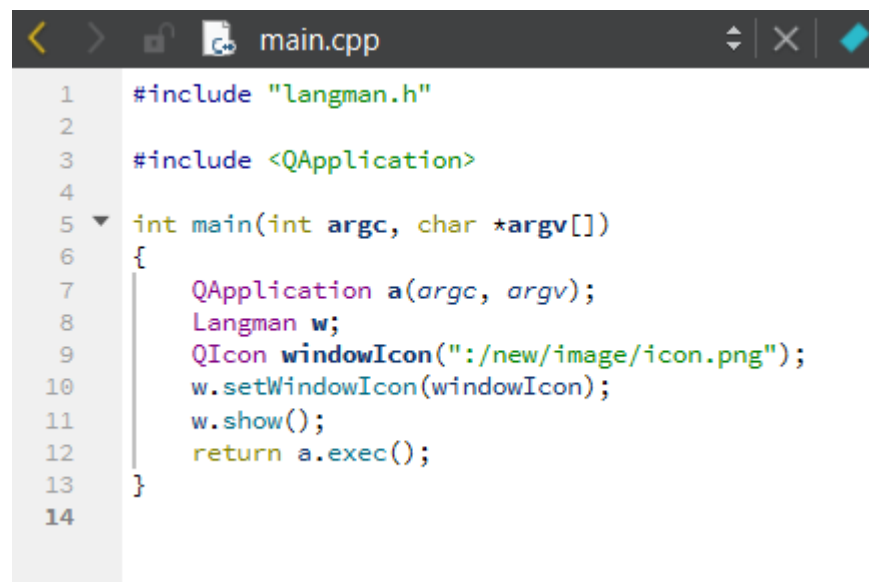
```

## word.cpp

```
word.cpp Letter::Letter(QWidg
1  #include "word.h"
2  // #include <fstream>
3  // #include <iostream>
4  #include <QJsonDocument>
5  #include <QJsonObject>
6  #include <QFile>
7
8  Letter::Letter(QWidget *parent):QLabel(parent)
9  {
10     |   isItTrue = false;
11
12 }
13
14 void Letter::fileReadJson()
15 {
16     QFile file(":/new/files/info.json");
17     if (file.open(QIODevice::ReadOnly | QIODevice::Text))
18     {
19         QByteArray data = file.readAll();
20         file.close();
21
22         QJsonDocument doc = QJsonDocument::fromJson(data);
23         if (!doc.isNull()){
24             jsonObject = doc.object();
25             qDebug() << "Json";
26             keys = jsonObject.keys();
27         }
28         else
29         { qDebug() << "JSON dosyası ayrıştırılmadı.";}
30     }
31     else
32     {
33         qDebug() << "JSON dosyası açılmadı.";
34     }
35 }
```

```
34     }
35 }
36
37 QStringList Letter::randomLang()
38 {
39     QStringList questionInfo;
40     int control = 1; //Aynı kelime tekrarı
41
42     while(control){
43
44         if(keys.size() != 0)
45         {
46             qDebug() << "Random" << keys.size();
47             int randomIndex = rand() % keys.size();
48             randomKey = keys[randomIndex];
49             qDebug() << "Key" << keys.size();
50             if(keys.contains(randomKey))
51             {
52                 control = 0;
53                 QJsonObject info = jsonObject[randomKey].toObject();
54                 int id = info["id"].toInt();
55                 QString clue = info["clue"].toString();
56                 QString infoText = info["info"].toString();
57
58                 questionInfo.append(randomKey);
59                 questionInfo.append(clue);
60                 questionInfo.append(infoText);
61
62                 qDebug() << "Key: " << randomKey;
63                 qDebug() << "ID: " << id;
64                 qDebug() << "Clue: " << clue;
65                 qDebug() << "Info: " << infoText;
66                 keys.removeOne(randomKey);
67                 qDebug() << "SONRAKİ KEY" << keys.size();
68             }
69         }
70         else if(keys.size() == 0){ //keys.size() = 0 ise oyun kazanılmıştır.
71             qDebug() << "Tebrikler Kazandınız";
72             questionInfo.append("KAZANDINIIZ!!");
73             return questionInfo;
74         }
75     }
76     return questionInfo;
77 }
78 }
```

## main.cpp



```
1  #include "langman.h"
2
3  #include <QApplication>
4
5  int main(int argc, char *argv[])
6  {
7      QApplication a(argc, argv);
8      Langman w;
9      QIcon windowIcon(":/new/image/icon.png");
10     w.setWindowIcon(windowIcon);
11     w.show();
12     return a.exec();
13 }
14
```

## KAYNAKÇA

- [1] <https://cplusplus.com/info/history/>
- [2] <https://www.geeksforgeeks.org/history-of-c/>
- [3] <https://developingthefuture.net/compilation-process-and-jit-compiler/>
- [4] <https://www.geeksforgeeks.org/cpp-variables/>
- [5] <https://www.baeldung.com/cs/orthogonalitycsprogramming-languages-software-databases>
- [6] [https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming#uses of c with realworld applications](https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming#uses_of_c_with_realworld_applications)
- [7] <https://www.geeksforgeeks.org/inheritance-in-c/>
- [8] [https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming#what is c used for](https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming#what_is_c_used_for)
- [9] [https://en.wikipedia.org/wiki/Category:Free software programmed in C%2B%2B](https://en.wikipedia.org/wiki/Category:Free_software_programmed_in_C%2B%2B)
- [10] <https://www.simplilearn.com/c-plus-plus-programming-for-beginners-article>
- [11] <https://www.geeksforgeeks.org/top-10-best-embedded-systems-programming-languages>
- [12] <https://www.geeksforgeeks.org/dynamic-binding-in-cpp/>
- [13] <https://learnloner.com/static-and-dynamic-binding/#:~:text=There%20are%20two%20types%20of,the%20function%20implementation%20is%20determined.>
- [14] [Concepts of Programming Languages 11th Ed](#)