

---

# C++

## 1. C++ Nedir?

C++, 1979'da Bell Labs'da Bjarne Stroustrup tarafından geliştirilen orta düzey bir programlama dili olarak geliştirilmiştir.

C++ ; Windows, Mac OS ve çeşitli UNIX sürümleri gibi çeşitli platformlarda çalışır.

C++ statik olarak geliştirilmiş, genel amaçlı büyük küçük harf duyarlı prosedürel ve nesne yönelimli programlamayı destekleyen bir dildir.

C++ yüksek ve düşük seviyeli programlama dilleri özelliklerini barındırdığı için orta seviyeli bir dildir.

C++, C dili üzerine geliştirildiği için C dilinin bir kalıtımıdır.

## 2. C++ Program Yapısı

Her programlama dilinde olduğu gibi Merhaba Dünya yazısı ile dilin yapısını inceleyelim.

```
#include <iostream>
using namespace std;

// main() main programı çalıştıran ana metottur.
int main() {
    cout << "Merhaba Dünya"; //Ekranı merhaba dünya yazdırılır
    return 0;
}
```

Aşağıdaki örnekte de değişken tanımlama ve bu değişkenlere değer atama işlemini görebiliriz.

```
#include <iostream>
using namespace std;

// Değişken tanımlama
extern int a, b;
extern float f;
```

```
int main () {  
    // degisken tanimlama:  
    int a, b;  
    float f;  
  
    // deger atama  
    a = 10;  
    b = 20;  
  
    cout <<"toplam :" << a + b << endl ;  
  
    f = 50.0/3.0;  
    cout << "ondalik :" << f << endl ;  
  
    return 0;  
}
```

Değişken atama işlemlerini fonksiyon içerisinde de yapabiliriz.

```
#include <iostream>  
using namespace std;  
  
// function beyani  
int ornek();  
int main() {  
  
    // function cagirma  
    int i = ornek();  
}  
  
// function tanimlama  
int ornek() {  
    return 125;  
}
```

C++ programlama dilinde if, if else ve switch-case karar yapıları kullanılmaktadır.

If else örneği aşağıdaki gibidir.

```
#include <iostream>  
using namespace std;  
  
int main () {  
    // degisken tanimlama
```

```

int su_sicaklik = 110;

// koşul kontrolü
if( su_sicaklik < 0 ) {
    // algoritmaornekleri
    cout << "su buz halindedir" << endl;
} else if(su_sicaklik <100) {
    //
    cout << "su sivi haldedir" << endl;
}
else
{
    cout << "su gaz haldedir" << endl;
}

cout << "su sicakligi : " << su_sicaklik << endl;

return 0;
}

```

For döngüsünün örneği aşağıdaki şekildedir.

```

#include <iostream>
using namespace std;
int main (){
    int toplam = 0;
    for (int i = 5; i <= 1000; i++){
        if(i%2 == 0 || i%5 == 0)
            toplam += i;
    }
    cout << "Toplam" << toplam << endl;
    return 0;
}

```

C++ fonksiyon kullanımı bir fonksiyon yada diğer adı ile işlev, bir görevi yerine getiren bir ifadeler topluluğudur.

Her C ++ programında adı main() olan en az bir fonksiyon bulunur ve tüm diğer küçük fonksiyonlar bu fonksiyona bağlanarak çalışır.

C++ da fonksiyon yapısı aşağıdaki şekildedir.

```

dönüş_tipi fonksiyon_adı ( parametre listesi ) {

```

```
fonksiyon gövdesi / yazılacak kodlar  
}
```

Fonksiyon kullanımıyla ilgili örnek kod ise şu şekilde olmaktadır:

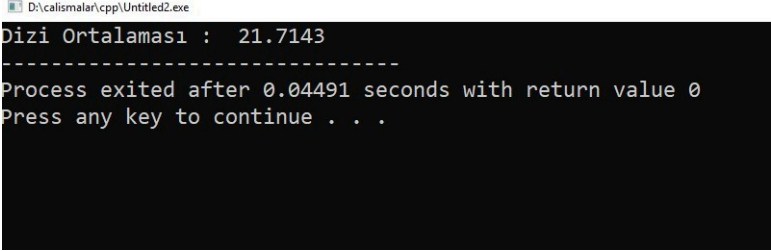
```
#include <iostream>  
using namespace std;  
  
// fonksiyon bildirimi  
int max(int num1, int num2);  
  
int main () {  
    // yerel değişken  
    int a = 10;  
    int b = 20;  
    int sonuc;  
  
    // fonksiyon çağırma  
    sonuc= max(a, b);  
    cout << "İki sayıdan büyük olan deger : " << sonuc<< endl;  
  
    return 0;  
}  
  
// fonksiyon tanımlama  
int max(int num1, int num2) {  
    // yerel değişken  
    int s;  
  
    if (num1 > num2)  
        s= num1;  
    else  
        s= num2;  
  
    return s;  
}
```

C++, aynı tipteki sabit boyutlu bir sıralı array koleksiyonunu saklayan bir veri yapısı dizisini sağlar. Bir veri topluluğunun depolamak için bir dizi kullanılır, ancak bir diziyi aynı türden değişkenlerin bir koleksiyonu olarak düşünmek genellikle daha yararlıdır.

Dizi kullanımıyla alakalı örnek kod aşağıdaki gösterilmiştir.

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL,"Turkish"); //Türkçe karakter
    int sayilar[5];
    for(int i=0;i<5;i++)
    {
        cout<<i+1<<". Sayı : ";
        cin>>sayilar[i];
    }
}
```

Ekran Çıktısı:



D:\calismalar\cpp\Untitled2.exe

Dizi Ortalaması : 21.7143

-----

Process exited after 0.04491 seconds with return value 0

Press any key to continue . . .

Dizi içindeki En Büyük ve En Küçük Değerli sayıları bularak ekranda gösteren C++ kodu da aşağıda verilmiştir.

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL,"Turkish"); //Türkçe karakter
    int sayilar[]={5,10,15,13,15,22,42};
    int buyuk=sayilar[0];
    int kucuk=sayilar[0];
    int n = sizeof(sayilar) / sizeof(sayilar[0]);
    for(int i = 0; i < n; i++)
    {
        if(sayilar[i]>buyuk);
        {
            buyuk=sayilar[i];
        }
        if(sayilar[i]<kucuk)
```

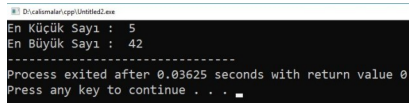
```

{
    kucuk=sayilar[i];
}
}

cout<<"En Küçük Sayı : "<<kucuk<<endl;
cout<<"En Büyük Sayı : "<<buyuk;
}

```

Ekran Çıktısı:



```

En Küçük Sayı : 5
En Büyük Sayı : 42
-----
Process exited after 0.03625 seconds with return value 0
Press any key to continue . . .

```

C++ bize iki tip string yapısı sunmaktadır.

1. Birincisi C stili karakter dizeleri
2. Diğer ise C++'da standart olarak tanımlı string sınıfıdır.

C stili karakter dizesi C dilinde var olan ve C++ içinde desteklenmeye devam eden '\0' boş karakterle sonlanan tek boyutlu bir karakter dizisidir. Karakter dizleri oluşturulduğunda dize sonuna konulmasa bile program tarafından bu sonlandırma işareti konulacaktır.

```

#include <iostream>

using namespace std;

int main () {

    char selam[6] = {'S', 'e', 'l', 'a', 'm', '\0'};

    cout << "Selam Mesajı: ";
    cout << selam<< endl;

    return 0;
}

```

C++ string fonksiyonları aşağıdaki tabloda verilmiştir.

Sıra	Fonksiyon ve Tanımı
1	<code>strcpy(s1, s2);</code> s2'yi s1 içine kopyalar
2	<code>strcat(s1, s2);</code> s2 dizisini s1 s1 dizisinin sonuna ekler.
3	<code>strlen(s1);</code> s1 dizisinin uzunluğunu getirir.
4	<code>strcmp(s1, s2);</code> s1 ile s2 dizisini karşılaştırır. s1 ie s2 aynı ise 0, s1<s2 durumda sıfırdan küçük bir değer s1>s2 durumda sıfırdan büyük bir değer döndürür.
5	<code>strchr(s1, ch);</code> s1(string) bulunan ch karakterinin pointerını döndürür. Aranana ch yoksa null döndürür. Aranana değer adres olduğu için dönen değer pointer olarak alınması gerekir.
6	<code>strstr(s1, s2);</code> s1 içinde s2 değerinin geçtiği ilk adresi döndürür.

Aşağıdaki örnek, yukarıda bahsedilen işlevlerin bir kısmını kullanır.

```
#include <iostream>
#include <cstring>

using namespace std;

int main () {

    char str1[10] = "Hello";
    char str2[10] = "world";
    char str3[10];
    int len ;

    // copy str1 into str3
    strcpy( str3, str1);
```

```
cout << "strcpy( str3, str1) : " << str3 << endl;

// concatenates str1 and str2
strcat( str1, str2);
cout << "strcat( str1, str2): " << str1 << endl;

// total length of str1 after concatenation
len = strlen(str1);
cout << "strlen(str1) : " << len << endl;

return 0;
}
```

C++ dili C dilinden türemesine rağmen en büyük farklı nesne yönelimli programlamaya izin vermesidir.

NYP tabanlı programlamada yazılımların daha anlaşılır olması için her şeyin bir nesne olduğu düşünülür.

NYP tabanlı programlama kalıtım, kapsülleme, çok biçimlilik gibi çeşitli özelliklere sahip olduğundan geliştirmeyi kolay ve güvenli hale getirir.

Aşağıda Kisi adında bir sınıf oluşturulmuştur.

```
#include <iostream>

using namespace std;

class Kisi {
public:
    string adi;
    string soyadi;
};

int main() {
    Kisi k1;
    k1.adi = "Deniz";
    k1.soyadi = "Yağmur";

    cout << "k1 adi: " << k1.adi << endl;
    cout << "k1 soyadi: " << k1.soyadi << endl;
    return 0;
}
```



Aşağıdaki örnekte class yapısında yer alan özellik, metot, kurucu metot, yıkıcı metot, erişim belirleyici, this anahtar kelimesi, sınıf sabiti ve pointer kullanımını içeren örnek yer almaktadır.

```
#include <iostream>

using namespace std;

class Kisi {
private:
    string adi;
protected:
    string soyadi;
public:
    static int nesneSayisi;
    Kisi(string a, string s) {
        cout << "Nesne olusturuluyor." << endl;
        this->adi = a;
        (*this).soyadi = s;
        Kisi::nesneSayisi++;
    }
    string yazdir() {
        return this->adi + " " + this->soyadi;
    }
    ~Kisi() {
        cout << "Nesne siliniyor." << endl;
    }
};

int Kisi::nesneSayisi = 0;

int main(){
    Kisi *p;
    Kisi k1("Deniz", "Yağmur");
    Kisi k2("Mehtap", "Yağmur");

    p = &k2;

    cout << k1.yazdir() << endl;
    cout << p->yazdir() << endl;

    cout << "Kopya sayisi : " << Kisi::nesneSayisi << endl;

    return 0;
```

```
}
```

NYP ile geliřtirmede kullanılan en önemli özellik kalıttır.

Kalıtım daha önceden oluşturulan sınıflara ait özellik ve metotların bir sınıfa aktarılmasıdır.

Ařağıdaki örnekte daha önceden oluşturulan Kisi sınıfında kalıtım alınmıřtır.

```
class Ogrenci : public Kisi {  
public:  
    string okul;  
    Ogrenci(string a, string s, string o):Kisi(a, s), okul(o) {};  
};
```

NYP ile kullanılan bir diğeri özellik ise çok biçimlilik.

Çok biçimlilik temel olarak taban sınıfın referansı ile türetilen sınıfa erişmektir.

Ařağıda çok biçimliliğin kullanımıyla ilgili örnek yer almaktadır.

```
Kisi *p;  
Ogrenci o1("Rabia", "SEZER", "ABC");  
p = &o1;  
  
cout << p->yazdir() << endl;
```

C++ programlama dilini öğrenmek için [youtube](https://youtu.be/naWZ2xVuF30)<sup>1</sup> daki programlama derslerinden yararlanabilirsiniz.

Youtube videosu:

<sup>1</sup> <https://youtu.be/naWZ2xVuF30>