

C++'a Giriş

- Program yapısı
- Değişkenler, veri türleri
- Sabitler
- Operatörler
- Temel Girdi/Çıktı (Input/Output, I/O)

Program Yapısı

- Hello World

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

int main ()
{
    cout << "Hello World!";
    return 0;
}
```

Program Yapısı

- `#include <iostream>`

Compiler'a iostream dosyasını kullanmasını söylüyor.

- `using namespace std;`

Her programın barındırdığı bir satır

- `int main()`

Ana programımız. `int` kodun tamsayı değer vereceğini, `()` ise `main`'in bir fonksiyon olduğunu söylüyor. `main()` kodun herhangi bir yerinde olabilir, öncesi veya sonrasında komutlar olabilir ama her zaman ilk önce çalıştırılır.

`main()`'i her zaman `{ }` takip eder ve fonksiyonun gövdesini belirtir.

Program Yapısı

- `cout << "Hello World" ;`

C++'in standard output komutu.

“” içindeki bilgiler ekrana yazdırılır.

Compiler cout'u nereden biliyor? eklediğimiz iostream dosyasından.

- `return 0;`

`int main()` tanımlamasını hatırlayın;

fonksiyonumuz tamsayı değer vermeli.

Genelde `main()` bu şekilde bitiriliyor.

Program Yapısı

- Programa biraz eklemeler yapalım.

```
/* my second program in C++  
   with more comments */  
  
#include <iostream>  
using namespace std;  
  
int main ()  
{  
    cout << "Hello World! ";    // prints Hello  
World!  
    cout << "I'm a C++ program"; // prints I'm a  
C++ program  
    return 0;  
}
```

Program Yapısı

- Yorumlar

// sonrası bütün karakterler yorum olarak algılanır. (tek satır)

/* */ arası bütün karakterler yorum olarak algılanır. (pek çok satır)

Program Yapısı

- `cout << "Hello World! I'm a C++ program";`
- `cout << "Hello World!"; cout << "I'm a C++ program";`
- `cout << "Hello World!";`
`cout << "I'm a C++ program";`
- Üstteki komutların hepsi aynı işi yapıyor. Önemli olan ; karakterini doğru kullanmak.
- **ÖNEMLİ:** bütün **C++** komutları ; ile biter.

Değişkenler ve Veri Türleri

- **Değişkenler:** $a=3$, $b=4$... matematikle aynı mantık. Her değişken bilgisayar hafızasında yer kaplıyor.
- a, b, \dots değişken isimleri. **ÖNEMLİ:** karakter, rakam veya `_` kullanılabilir. Büyük, küçük harf ayrımı var.
- C++'da rezerve isimler var dikkat!
- **Veri türleri:** Bilgisayara ne tür birşeyle uğraştığını söylememiz gerekiyor. Pek çok veri türü var; her birinin hafızada kapladığı boyut farklı.

Değişkenler ve Veri Türleri

- Çeşitli veri türleri ve ayrıntılı bilgileri:

represented with each one:

Name	Description	Size*	Range*
char	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	Boolean value. It can take one of two values: true or false.	1byte	true or false
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)

- Veri boyutları sisteme göre değişebilir.

Değişkenler ve Veri Türleri

- Program hangi veri türünü kullanacağını nasıl bilecek?
Biz söyleceğiz!
- `int a;` (birden fazla `int a,b,c;`)
- `double sayım;`
- `float utku88;`
- `long int x_y;`
- Eşdeğer tanımlamalar var:
`short a;` ve `short int a;` aynı veri türünü tanımlıyor vb.

Değişkenler ve Veri Türleri

- Değişkeni isimlerdirdik, türünü belirttik şimdi de bir değer atayalım.
- tür isim = değer; veya tür isim (değer);
- Yani int a=5; veya int a(5);
- **Son bir tür: String**

```
// my first string
#include <iostream>
#include <string>
using namespace std;

int main ()
{
    string mystring = "This is a string";
    cout << mystring;
    return 0;
}
```

- Kullanmak için `#include <string>`

Sabitler

- Kod örneğine bakalım.

```
// defined constants: calculate circumference
#include <iostream>
using namespace std;

#define PI 3.14159
#define NEWLINE '\n'

int main ()
{
    double r=5.0;           // radius
    double circle;

    circle = 2 * PI * r;
    cout << circle;
    cout << NEWLINE;

    return 0;
}
```

- #define isim değer (compiler kodu olduğu için ; yok)
- veya **const** int a=4;

Operatörler

- **Eşleme operatörü (=)**

a=5; a=b; a=2+(b=5); a=b=c=5;

```
int main ()  
{  
    int a, b;  
    a = 10;  
    b = 4;  
    a = b;  
    b = 7;
```

- sonuç: a=4, b=7.

Operatörler

- **Aritmetik (+, -, *, /, %:kalan)**
- **Birleşik operatörler (+=, -=, *=, /=, %=)**

`a+=b; : a=a+b;`

- **Arttırma, azaltma (++ , --)**

`a++; : a=a+1; a--; : a=a-1;`

NOT: `a=++b` ile `a=b++` farklı şeyler.

Example 1	Example 2
<pre>B=3; A=++B; // A contains 4, B contains 4</pre>	<pre>B=3; A=B++; // A contains 3, B contains 4</pre>

Operatörler

- İlişki ve eşitlik (==, !=, >, <, >=, <=)

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

```
(7 == 5) // evaluates to false
(5 > 4) // evaluates to true.
(3 != 2) // evaluates to true.
(6 >= 6) // evaluates to true.
(5 < 5) // evaluates to false
```

```
(a == 5) // evaluates to false since a is not equal to 5.
(a*b >= c) // evaluates to true since (2*3 >= 6) is true.
(b+4 > a*c) // evaluates to false since (3+4 > 2*6) is false.
((b=2) == a) // evaluates to true.
```

- If kontrol grubunu kullanırken çok işe yarayacaklar.

Operatörler

- Mantık (!, &&, ||)

!: NOT

&&: AND

||: OR

&& OPERATOR

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

|| OPERATOR

a	b	a b
true	true	true
true	false	true
false	true	true
false	false	false

```
!(5 == 5) // evaluates to false because the expression at its right (5 == 5) is true.
!(6 <= 4) // evaluates to true because (6 <= 4) would be false.
!true     // evaluates to false
!false    // evaluates to true.
```


Temel Girdi/Çıktı

- Hello World deki **cout <<** bir çıktı örneği.

```
cout << "Output sentence"; // prints Output sentence on screen
cout << 120;                // prints number 120 on screen
cout << x;                  // prints the content of x on screen
```

```
cout << "Hello"; // prints Hello
cout << Hello;   // prints the content of Hello variable
```

```
cout << "Hello, I am " << age << " years old and my zipcode is " << zipcode;
```

```
Hello, I am 24 years old and my zipcode is 90064
```

- << kullanımına güzel bir örnek.

Temel Girdi/Çıktı

```
cout << "First sentence.\n ";  
cout << "Second sentence.\nThird sentence.";
```

- First sentence
Second sentence
Third sentence

```
cout << "First sentence." << endl;  
cout << "Second sentence." << endl;
```

- First sentence
Second sentence
- \n ile << endl işlevi aynı. Bir satır aşağı.

Temel Girdi/Çıktı

- cin >> girdi komutu
- int a;
cin >> a;

```
// i/o example
#include <iostream>
using namespace std;

int main ()
{
    int i;
    cout << "Please enter an integer value: ";
    cin >> i;
    cout << "The value you entered is " << i;
    cout << " and its double is " << i*2 << ".\n";
    return 0;
}
```