# BM101 Algoritma ve Programlama I

# C++ Dilinin Temelleri

# İçerik



- 1. Giriş
- 2. Çıktı Yazdırma cout
- 3. Girdi Alma cin
- 4. Değişken Tanımı
- 5. Veri Tipleri
- 6. İsim Sabitleri
- 7. Atama Komutu
- 8. Bloklar

Çözümlü Sorular

### Hedefler



- cout komutu ile çıktı yazdırma
- cin komutu ile girdi okuma
- Geçerli değişken tanımı yapma
- Geçersiz değişkenleri ayırt etme
- Farklı veri tipleri ile değişken tanımıyapma
- İsim sabiti tanımlama
- Değişkenlerde atama komutu ile veri saklama
- Sıralanmış veri tipi tanımlama
- Blok kavramını anlatma
- İsim alanları oluşturma

## 1. Giriş

- C++ ile yazılmış bir program, bir veya birden fazla fonksiyondan oluşur.
- Fonksiyonlar bir görevi yapan komutlar grubudur.
- Her C++ programı mutlaka main() fonksiyonu içermelidir çünkü programın çalıştırılması bu fonksiyon ile başlar.
- Yazdığımız her komut, noktalı virgül (;) ile sonlandırılmalıdır.

#### // ilk programımız

```
1 #include <iostream>
2 using namespace std;
3  // ilk programımız
4 int main()
5 {
6    cout << "Merhaba C++ !";
7    return 0;
8 }</pre>
```

#### #include <iostream>

- Girdi/çıktı komutlarını içeren hazır kütüphanelere erişimi sağlar.
- Bu kütüphanede bulunan fonksiyonlar kullanılarak, kullanıcıdan veri okunabilir ve sonuçlar ekrana yazdırılabilir.

#### using namespace std;

• Bu komut, programa dahil edilen kütüphaneleri ayırt etmek için kullanılmaktadır.

#### // İlk programımız

- Açıklama satırı (comment)
- Açıklama satırları programın akışını etkilemez ve sadece program hakkında bilgi vermek için kullanılır.
- Açıklama satırı yazmak için /\*
   \*/ işaretleri de kullanılabilir.
- Örn. /\* İlk programımız\*/

```
int main()
{
    ...
}
```

- main() bir fonksiyondur
- int tamsayı (integer) tipinde bir değerdöndürür.
- Her fonksiyonun başı ve sonu { } ayraçları ile belirtilir.

Ekrana tırnak işaretleri içinde yer alan cümleyi yazar.



### return 0;

- main() fonksiyonunu bitirir.
- Programın başarılı bir şekilde bittiğini gösteren 0 değerini işletim sistemine döndürür.

### 2. Çıktı Yazdırma - cout

- Standart çıktı nesnesi olan cout, <iostream> kütüphanesinde bir veri yolu nesnesi (stream object) olarak tanımlanmıştır.
- Buradaki veri yolunu, çıktı ekranı ile program arasında bir yol olarak düşünebiliriz. Bu yola aktarılan her veri çıktı ekranına yansıtılacaktır.
- cout nesnesini takip eden << operatörü ise *veri yolu ekleme* (stream insertion) operatörü olarak adlandırılır. Bu operatörün sağ tarafında yazılanlar doğrudan ekrana yansıtılır.

## 2. Çıktı Yazdırma - cout...

#### Çıktı

$$7+3 = 10$$

## 2. Çıktı Yazdırma - cout...

```
#include <iostream>
using namespace std;
int main()
    cout << "7+3=" << 7+3 << endl;
    return 0;
            //Bilgileri tek bir cout
            komutuyla da
            yazdırabiliriz.
```

#### Çıktı

7+3 = 10

## 2. Çıktı Yazdırma - cout...

```
#include <iostream>
using namespace std;
int main()
{
    cout << "7+3"<<endl<<"="<<endl<< 7+3<<endl;
    return 0;
}</pre>
```

#### Çıktı

7+3 = 10

### 3. Girdi Alma - cin

- Standart girdi nesnesi olan cin, cout gibi <iostream> kütüphanesindedir.
- cin'den sonra yazdığımız >> operatörü ise veri yolu elde etme (stream extraction) operatörü olarak adlandırılır.
- Bu operatörün sağ tarafında yazılan değişken veya değişkenlerin içine, kullanıcı tarafından girilen veriler okunur.

### 3. Girdi Alma - cin...

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

int main()

{
   int yil;
   cout << "Hangi yilda dogdunuz? ";
   cin>>yil;
   cout<<"Siz "<<2014-yil<<" yasindasiniz.";
   return 0;
}</pre>
```

#### Çıktı

```
Hangi yilda dogdunuz? 1990
Siz 24 yasindasiniz.
```

### 3. Girdi Alma - cin...

- Bir programda verileri bellekte tutmak, saklanan verileri işlemlerde kullanmak için bellek hücreleri kullanılır.
- Bu bellek hücrelerinde saklanan veriler değişebileceği için bellek hücrelerine verilen isimlere değişken (variable) adı verilir.

Örnek: int yil;

• Bu komutla y i I adlı değişkeni i n t (tamsayı) tipinde tanımlamış oluyoruz.

### 3. Girdi Alma - cin...

• Bu komut y i I değişkenine kullanıcı taranndan klavye kullanarak girilen tamsayı değerini atar.

• Bu komut ile hem i s i m hem de y i I değişkenlerine kullanıcıdan veri okunacaktır.

- Değişken isimleri anlamlı ve C++ dilinde kullanılan anahtar sözcüklerden farklı olmalı ve aşağıdaki kurallara uygun bir şekildeseçilmelidir:
  - İlk harf alfabetik (a-z, A-Z) veya alt çizgi ( \_ ) karakteri olmalıdır.
  - Diğer harfler a-z, A-Z, 0-9 veya \_ karakterleri olabilir.
  - Büyük harf/küçük harf ayrımı vardır (numara, Numara, NUMARA birbirlerinden farklı değişkenlerdir).

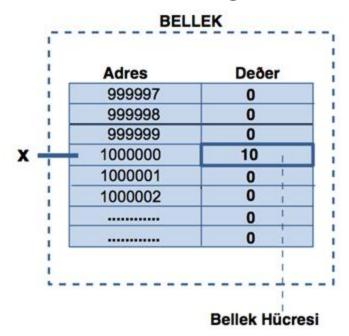
#### Geçerli ve Geçersiz Değişken İsimlerine Örnekler

Değişken ismi	Geçerli/geçersiz	Açıklama
hangiGun	Geçerli	
2sayi	Geçersiz	rakam ile başlanmaz.
Mart2001	Geçerli	
_sinif_mevcudu	Geçerli	
sinif#	Geçersiz	# gibi özel karakterler kullanılamaz.

## VeriTipi değiskenİsmi;

- Örnek: int x;
- Bu durumda bellekte x için i n t (tamsayı) boyutunda bir yer açılır.

- Örnek: int x=10;
- x'in bellek hücresine 10 değeri atanır.



x Değişkeninin Bellek Hücresindeki Görünümü

```
#include <iostream>
using namespace std;
int main()
{
   int x=2;
   cout << x;
   int y=4;
   cout << y;
   return 0;
}</pre>
```



Değişkenleri ilk kez kullanmadan önce mutlaka tanımlamamız gerektiğini unutmayalım.

# 5. Veri Tipleri

#### Temel Veri Tipleri

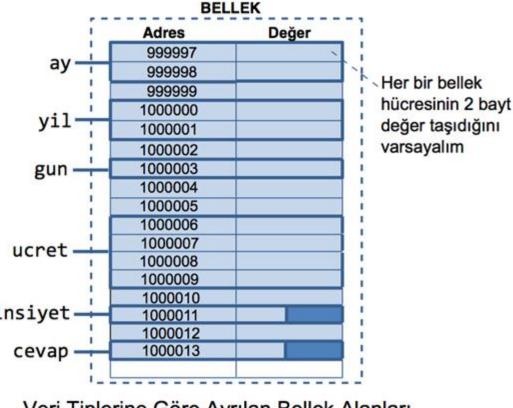
Veri Tipi	Açıklama	Bayt	Aralık
short	Kısa tamsayı	2 bayt	-32768 ↔ 32768
int	Tamsayı	4 bayt	-2,147,483,648 ↔
			+2,147,483,647
long	Uzun tamsayı	4 bayt	-2,147,483,648 ↔
2000		1,50	+2,147,483,647
unsigned int	İşaretsiz tamsayı	4 bayt	0 ↔ 4,294,967,295
float	Reel sayı	4 bayt	-3.4x10 <sup>-38</sup> ↔ 3.4x10 <sup>38</sup>
double	Çift duyarlı reel sayı	8 bayt	-1.7x10 <sup>-308</sup> ↔ 1.7x10 <sup>308</sup>
long double	Uzun çift duyarlı reel sayı	10 bayt	-3.4x10 <sup>-4932</sup> ↔ 1.1x10 <sup>4832</sup>
char	Karakter	1 bayt	
bool	Mantiksal	1 bayt	true ↔ false

### 5. Veri Tipleri...

- Tamsayı (int): 2 -32 0
- Reel sayı (float, double): 2.35 -0.4567 0.0
- Karakter (char): 'A' '!' '9'
- Mantiksal (bool): true false
- Örnek:

## 5. Veri Tipleri...

```
// tamsayi tipinde ay
int ay;
// pozitif tamsayi tipinde yil
unsigned yil;
// kısa tamsayi tipinde gun
short gun;
// çift duyarlı reel sayı tipinde ucret cinsiyet-
double ucret;
// karakter tipinde cinsiyet
char cinsiyet;
// mantiksal veri tipinde cevap
bool cevap;
```



Veri Tiplerine Göre Ayrılan Bellek Alanları

## 5. Veri Tipleri...

 string tipinde tanımlanmış değişkenler birden fazla karakteri tutabilirler.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string kelime="Merhaba";
    cout << kelime;
    return 0;
}</pre>
```

#### Çıktı

Merhaba

### 6. İsim Sabitleri

• *isim sabiti* (constant) olarak tanımlanan bu değişkenin değeri, program akışında bir daha değiştirilemez.

```
const veriTipi sabitİsmi = ifade;
```

#### Örnek:

```
const int sinif = 50;
const float pi = 3.14;
```

### 6. İsim Sabitleri...

#### Çıktı

```
Yari capi giriniz:2
Dairenin cevresi = 12.56
```

### 7. Atama Komutu

 Değişkenlerin sakladıkları verileri değiştirmek için atama komutu (assignment statement) kullanılır.

Örnek:

$$a = 10;$$

### 7. Atama Komutu...

#### Temel Aritmetik Operatörleri

Operatör	Anlam	Örnek
+	Toplama	toplam = toplam + 20;
-	Çıkartma	ucret = maas - vergi;
*	Çarpma	toplamMaas = maas * 12;
/	Bölme	yarim = tam / 2;
%	Kalan	birlik = sayi % 10;
-	Tekli eksi	x = -5;

### 7. Atama Komutu...

- x= 7 / 2; // cevap: 3 (tamsayı bölme)
- y= 7.0 / 2.0 // cevap: 3.5 (reel sayı bölme)
- x = 14 % 3; // cevap: 2 (kalanı bulur)

#### Aritmetik Operatörlerin Öncelik Sırası

Operatör	Öncelik	Birleşme özelliği
Parantez ( )	Yüksek	İçten dışa
Tekli eksi (-)	Düşük	Sağdan sola
* / %		Soldan sağa
+ -	*	Soldan sağa

### 7. Atama Komutu...

```
x = 3 + 5 * 4 / 2 - (1 + 2); // öncelik sırasına göre önce () uygulanır x = 3 + \frac{5 * 4}{2} - 3; // öncelik sırasına göre * işlemi yapılır x = 3 + \frac{20}{2} - 3; // öncelik sırasına göre / işlemi yapılır x = \frac{3 + 10}{3} - 3; // birleşme özelliği ile önce + işlemi yapılır x = \frac{13 - 3}{3}; x = 10;
```

### 8. Bloklar

• {} ayraçları arasında yer alan komut dizisine *blok* (block) adı verilir.

```
{
  int x;
  cin>>x;
  cout<<x;
}</pre>
```

Her değişkenin bir kapsamı (scope) vardır.
 Değişkenlerin nerede tanımlandığı, bu değişkenlerin program içerisinde nerede kullanılabileceğini değişkenin kapsamı belirler.

### 8. Bloklar...

```
Blok3
int i;
                  Blok2
    char i;
               Blok1
         float i;
                                       3.5
         i=3.5;
       = 'A';
  = 5;
```

Bloklar ve Değişkenlerin Kapsamı

## Çözümlü Sorular

#### Soru

Aritmetik operatörler ve öncelik sıralarını düşünerek, aşağıdaki işlemlerin sonucunu hesaplayınız.



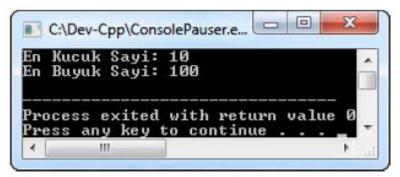
- **a.** 8
- **b.** 14
- c. 11
- d. 213
- e. -10

#### Soru

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

// Bu program iki degiskene baslangic
// degeri atar ve onlari ekranda
// goruntuler

int main() {
  int enaz=10;
  int encok=100;
  cout <<"En Kucuk Sayi: " << enaz << "\n";
  cout <<"En Buyuk Sayi: " << encok << "\n";
  return 0; }</pre>
```

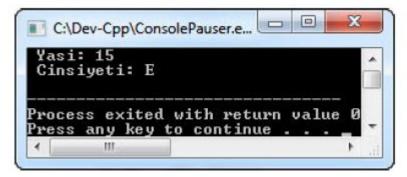


#### Soru

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

int const yas = 15;
char const cins = 'E';

int main() {
  cout << " Yasi: " << yas <<"\n";
  cout << " Cinsiyeti: " << cins <<"\n";
  return 0; }</pre>
```



#### Soru

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

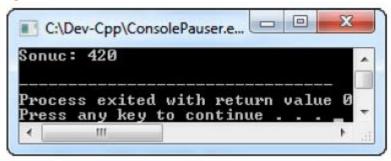
int main(){
  int sayi=9;
  float a,b,c;
  a=sayi/4;
  b=sayi/4.0;
  c=(float)sayi/4;
  cout << "a degeri= " << a << endl;
  cout << "b degeri= " << b << endl;
  cout << "c degeri= " << c << endl;
  return 0; }</pre>
```

#### Soru

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

// Bu program iki sayisal degeri
// toplayarak sonucu goruntuler

int main() {
int sayi1=120;
int sayi2=300;
int toplam;
toplam=sayi1+sayi2;
cout << "Sonuc: " << toplam << "\n";
return 0; }</pre>
```

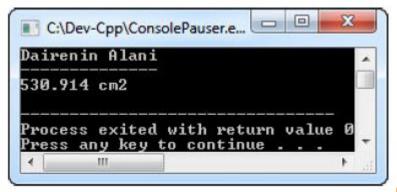


#### Soru

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

// Dairenin Alan1

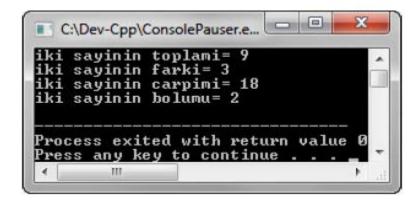
int main() {
  double pi=3.1415;
  double yaricap=13;
  double alan;
  cout << "Dairenin Alani" << "\n";
  cout << "-----" << "\n";
  alan=pi*yaricap*yaricap;
  cout << alan << " cm2" << "\n";
  return 0; }</pre>
```



#### Soru

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

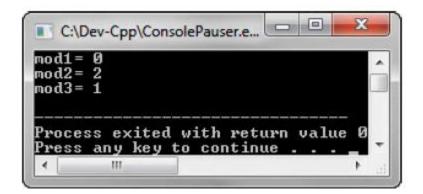
int main() {
  int sayi1=6,sayi2=3;
  cout << "iki sayinin toplami= "
  << sayi1+sayi2 << endl;
  cout << "iki sayinin farki= "
    << sayi1-sayi2 << endl;
  cout << "iki sayinin carpimi= "
    << sayi1*sayi2 << endl;
  cout << "iki sayinin bolumu= "
    << sayi1/sayi2 << endl;
  return 0; }</pre>
```



#### Soru

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

int main(){
  int x=8,y=4,z=3;
  int mod1=x%y;
  int mod2=x%z;
  int mod3=y%z;
  cout << "mod1= " << mod1 << end1;
  cout << "mod2= " << mod2 << end1;
  cout << "mod3= " << mod3 << end1;
  return 0; }</pre>
```



#### Soru

Kare ve dikdörtgenin alan hesabını yapan bir program verilmiş ve programda bazı kısımlar eksik bırakılmıştır. Örnek çıktıyı dikkate alarak programı tamamlayınız.

#### Örnek Çıktı

```
Karenin kenar uzunlugunu giriniz: 9
Dikdortgenin kisa kenar uzunlugunu giriniz: 3
Dikdortgenin uzun kenar uzunlugunu giriniz: 12
Karenin alani: 81
Dikdortgenin alani: 36
#include <iostream>
using namespace std;
int main()
     _____ kAlan, dAlan;
    _____ a, b;
   cout<<"Karenin kenar uzunlugunu giriniz:";</pre>
   cout<<"Dikdortgenin kisa kenar uzunlugunu giriniz:";
   cout<<"Dikdortgenin uzun kenar uzunlugunu giriniz:":
   dAlan = ;
   cout<<"Karenin alani:"<< ____ <<endl;
   cout<<"Dikdortgenin alani:"<< <<endl;
   return 0;
```

```
#include <iostream>
using namespace std;
int main()
    int kAlan, dAlan;
    int a, b;
    cout<<"Karenin kenar uzunlugunu giriniz:";
    cin>>a;
    kAlan = a*a;
    cout<<"Dikdortgenin kisa kenar uzunlugunu giriniz:";
    cin>>a;
    cout<<"Dikdortgenin uzun kenar uzunlugunu giriniz:";
    cin>>b;
    dAlan = a*b;
    cout<<"Karenin alani:"<<kAlan<<endl;</pre>
    cout<<"Dikdortgenin alani:"<<dAlan<<endl;</pre>
    return 0;
}
```

#### Soru

Üç basamaklı bir tamsayı okuyunuz ve bu sayının basamaklarının toplamını bulan bir program yazınız.

### Örnek Çıktı

3-basamaklı bir sayı giriniz: 428

428 sayısının basamak toplamı: 4+2+8=14



```
#include <iostream>
using namespace std;
int main()
{
    int b1,b2,b3,sayi;
    cout<<"3-basamakli bir sayi giriniz: ";
    cin>>sayi;
    b1=sayi/100;
    b2=sayi%100/10;
    b3=sayi%10;
    cout<<sayi<<" sayisinin basamak toplami:";
    cout<<b1<<"+"<<b2<<"+"<<b3<<"="<<b1+b2+b3<<endl;
    return 0;
}</pre>
```