# BM101 Algoritma ve Programlama I Fonksiyonlar-1

# İçerik

- Standart Kütüphane Fonksiyonları
- Kullanıcı Tarafından Tanımlanan Fonksiyonlar
  - Fonksiyon Tanımı
  - Fonksiyon Prototipi
  - void Fonksiyonları
  - Parametreli Fonksiyonlar
  - Varsayılan Parametreler
- Değişken Kapsamı
- Tekli Kapsam Çözme Operatörü ::
- Fonksiyon Yükleme
- Satıriçi Fonksiyonlar

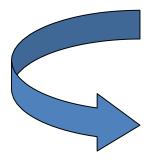
### Fonksiyonlar

- Fonksiyonlar bir görevi yapan komutlar grubudur.
- Fonksiyonlar uzun bir problemi daha küçük problemlere bölerek çözmek ve program yazımını basitleştirmek için kullanılır.
- Fonksiyonu bir kere yazıp programın farklı yerlerinden çağırmak mümkündür.

# Standart Kütüphane Fonksiyonları

- •Standart kütüphanelerde değişik amaçlar için kullanılabilen birçok fonksiyon tanımlanmıştır.
- •Örneğin, aşağıdaki fonksiyonları kullanabilmek için **cmath>** kütüphanesini #include komutu ile programa katmamız gerekir.

Matematiksel İfade	Fonksiyon
<b>3</b> 5	pow(3.0,5.0)
$\sqrt{16}$	sqrt(16.0)
-8	abs(-8)



#### ÇIKTI

3 uzeri 5= 243 16'nin kare koku= 4 -8'in mutlak degeri= 8

# Kullanıcı Tarafından Tanımlanan Fonksiyonlar

```
VeriTipi fonksiyonAdı (parametre listesi)
          Fonksiyon Adı
  Veri Tipi
                     Parametre Listesi
     int main ()
                             Fonksiyon Başlığı
          cout << "Merhaba C++!";
                                                 Fonksiyon
                                                 Gövdesi
          return 0;
```

```
#include <iostream>
_using namespace std;
                                        Fonksiyon
_i<u>nt_</u>f<u>ak</u>toriyel(<u>i</u>nt <u>x</u>) __
                                        Başlığı
    int f=1;
                                        Fonksiyon
    for (int i = 1; i < = x; i++)
                                        Gövdesi
      f *= i;
  _ return f;
main(){
    int sayi;
                                             Fonksiyon
    cout << "Sayi giriniz:";
                                             Çağırma
    cin>>sayi;
    cout<<sayi<<"!="<
    faktoriyel(sayi)<<endl;
                                        Sayi giriniz:4
                                        4!=24
```

```
#include <iostream>
 using namespace std;
 int faktr(intx)
    return f;
main(){
                                2
    cout<<sayi<<"!="<<faktr(sayi)<<endl;</pre>
5
```

```
#include <iostream>
using namespace std;
int fakt(int x)
                        // Faktoriyel hesaplayan fonksiyon
    int f=1;
    for (int i=1; i <= x; i++)
        f *= i;
    return f;
int main()
                                                Bir sayi giriniz:4
                                                 Sonuc=13
    int n, sonuc;
    cout<<"Bir sayi giriniz:";
    cin>>n;
             // Girilen n değerini kullanarak fonksiyon 4 kez çağırılır
    sonuc=(fakt(n)/4)*(6/fakt(n-1))+(fakt(n-2)/2)*(7/fakt(n-3));
    cout<<"Sonuc="<<sonuc<<endl;
    return 0;
```

## Fonksiyon Prototipi

- Kullandığımız her ismin (değişken, sabit, fonksiyon, vs.) kullanımından önce tanımlanması gerekir.
- Fonksiyonları önceden tanıtıp fonksiyon içeriğini istediğimiz yere, hatta başka bir dosyaya bile yazmamız, fonksiyon prototipi (function prototype) ile mümkün olabilmektedir

```
int fakt(int x); // veya
int fakt(int);
```

```
#include <iostream>
using namespace std;
int fakt(int);
                                     // Fonksiyon prototipi
int main()
  int n, sonuc;
  cout<<"Bir sayi giriniz:";</pre>
           // Girilen n değerini kullanarak fonksiyon 4 kez çağırılır
  cin>>n:
  sonuc=(fakt(n)/4)*(6/fakt(n-1))+(fakt(n-2)/2)*(7/fakt(n-3));
  cout<<"sonuc ="<<sonuc<<endl;
  return 0;
int fakt(int x)
                                      // Faktoriyel hesaplayan fonksiyon
    int f=1;
    for (int i = 1; i <= x; i++)
         f *= i:
    return f;
```

#### void Fonksiyonları

Bir fonksiyon değer döndürmüyorsa veya parametre içermiyorsa, bu durumu void yazarak belirtmemiz gerekir.

```
#include <iostream>
usingnamespace std;
void mesaj(void);
                           // Fonksiyon prototipi
int main()
    mesaj();
                           // Fonksiyon çağırımı
    return 0;
void mesaj(void)
                           // Mesaj yazan fonksiyon
    cout < < "Kullaniciyamesaj
    cout << "ileten bir fonksiyon" << endl;
                      Kullaniciya mesaj ileten bir fonksiyon
```

```
#include <iostream>
                                                              main
using namespace std;
                                                                      3.4
                                   // Fonksiyon prototipi
void buyuk(float,float);
int main()
                                                                       X
    float x,y;
                                                              buyuk
    cout<<"lki deger giriniz:";
                                    // İki sayı girilir
    cin>>x>>y;
                                                                      3.4
                                    //İki sayı fonksiyona yollanır
    buyuk(x, y);
                                                                       а
    return 0;
void buyuk(float a,float b)
                                    // İki sayının büyüğünü bulan fonksiyon
    cout<<"Sayilarin buyugu:";
    if (a>b)cout<<a<<endl; // İki sayı karşılaştırılır
         else cout<<b<<endl;
                                   // Büyüğü ekrana yazdırılır
```



#### **CIKTI**

Iki deger giriniz:3.4 2.1 Sayilarin buyugu:3.4

2.1

2.1

b

```
#include <iostream>
using namespace std;
double alan(void);
                                    // Fonksiyon prototipi
int main()
    double sonuc;
     sonuc=alan();
                                    // alan() fonksiyonu çağırılır, döndürdüğü değer sonuc'a atanır
     cout<<"Alan = "<<sonuc<<endl;</pre>
     return 0;
double alan (void) // En ve boyu okuyup dikdörtgenin alanını hesaplayan fonksiyon
     double en,boy;
     cout<<"Dikdortgenin en ve boyunu giriniz:";
     cin>>en>>boy;
                                   // En ve boyu okur
                                    // Alanı hesaplayıp döndürür
     return en*boy;
```



#### **CIKTI**

Dikdortgenin en ve boyunu giriniz: 2.2 3.3 Alan = 7.26

#### Parametreli Fonksiyonlar

- ·Çağırılan fonksiyonla çağıran fonksiyon arasında veri alışverişi *parametre* kullanılarak yapılır.
- •Çağıran fonksiyonda yer alan parametrelere *gerçek* parametre, çağırılan fonksiyonda yer alan parametrelere formal parametre ismi verilir.
- •Bu parametrelerin sayısının ve veri tipinin uyumlu olması gerekir.

```
void fon(int,float);
int main()
           int x=5;
           float y=3.6;
           fon(x.y);
                              Gerçek parametre
void fon(int a. float b)
                           → Formal parametre
                  Bu parametreler değer parametresi
                  (call by value) veya referans
                  parametresi (call by reference)
                  olarak kullanılabilir.
```

# Referans Değişkeni

- •Referans değişkenleri bir değişkene farklı bir isim vermek için kullanılır.
- Referans değişkenleri & (referans) operatörü ile tanımlanır.

Referans değişkenlerini tanımladığımızda mutlaka ilk değer atamasını yapmamız gerektiğini, bir başka deyişle hangi değişkenin referansı olacağını belirtmemiz gerektiğini unutmayalım.

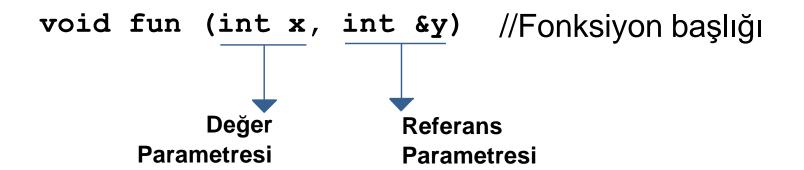
```
int &ref; // X geçersiz tanımlama
```

Bir referans değişkeni tek bir değişkenin referansı olabilir ve bu ilk atama sırasında belirlenir. Daha sonra bu referans değişkeni farklı bir değişkenle ilişkilendirilemez.

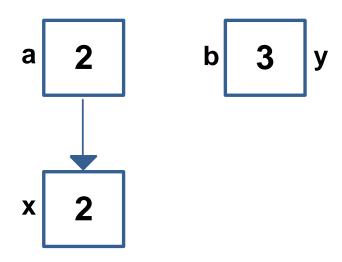
```
int x=5,y=6;
int &ref=x;
ref=y; // X Geçersiz atama
```

# Fonksiyon Parametreleri

- C++ dilinde parametreler iki şekildeyollanabilir.
- Değer Parametresi: Çağırılan fonksiyona gerçek parametre değerinin bir kopyası yollanır. Fonksiyonda formal parametre için bellekte bir yer açılır ve yollanan değer bu yere kopyalanır. Fonksiyon bitiminde yeni açılan yer yok olur ve gerçek parametrenin değeri etkilenmez.
- Referans Parametresi: Fonksiyona gerçek parametrenin referansı yollanır. Böylece fonksiyon başlığında yer alan formal parametre ve yollanan gerçek parametre bellekte aynı hücreyi paylaşır. Fonksiyonda formal parametreye yapılan herhangi bir değişiklik gerçek parametreye de yansır.



fun(a,b); // Fonksiyonçağırımı



```
#include <iostream>
using namespace std;
void surpriz (int a, int &b)
    a=5;
                                           10
    b=6;
    cout<<"a= "<<a<<" b= "<<b<<end l;
int main()
    int i = 10, j = 20;
    surpriz(i,j);
    cout << "i= "<<i<<" j= "<<j<<endl;
    return 0;
                                         a = 5 b = 6
```

# Varsayılan Parametreler

- Çağırılan fonksiyondaki formal parametreler ile fonksiyon çağırımında yer alan gerçek parametrelerin sayı ve tip olarak uyuşması gerekir.
- Çağıran fonksiyon, çağırdığı fonksiyona gerçekparametre yollamayacaksa, varsayılan parametreler (default parameter) kullanılır.
- Gerçek parametre yollanmadığında daha önce tanımlanmış değeri alan parametrelere varsayılan parametre denir.

```
#include <iostream>
using namespace std;
void fon (int x=1, int y=20)
                                         // Varsayılan değerlerle 2 formal parametre
     cout<<x<" "<<y<endl;
int main()
     fon(5,10);
                                          // 2 gerçek parametre ile fonksiyon çağırımı
     fon(5);
                                         // 1 gerçek parametre ile fonksiyon çağırımı
     fon();
                                         // Gerçek parametresiz fonksiyon çağırımı
     return 0;
```



**ÇIKTI**5 10
5 20
1 20

 Varsayılan parametreler parametre listesinin sonunda yer almalıdır.

```
void fon(int x=5, int y) // X Hata
```

 Varsayılan parametreleri fonksiyon prototipinde veya fonksiyon başlığında belirtebiliriz, ancak ikisinde birden belirtmememiz gerekir.

```
void fon(int, int=5);  // fonksiyon prototipi
....
void fon (int y, int x=5)  // fonksiyon başlığı
{....}
```