Bölüm 6
Fonksiyonlar

- Fonksiyon Tanımı
- Değer Döndürmeyen Fonksiyonlar
- Değer Döndüren Fonksiyonlar
- Çok Parametreli Fonksiyonlar
- Değişken Kapsamları
- Çok Fonksiyonlu Programlar

Fonksiyon Tanımı

Karmaşık ve uzun programları, küçük, basit ve belirli bir amacı olan program parçalarına bölebiliriz. Belirli bir işi yapan bu program parçalarına ***fonksiyon*** adı verilir. **`main`** ana fonksiyondur.

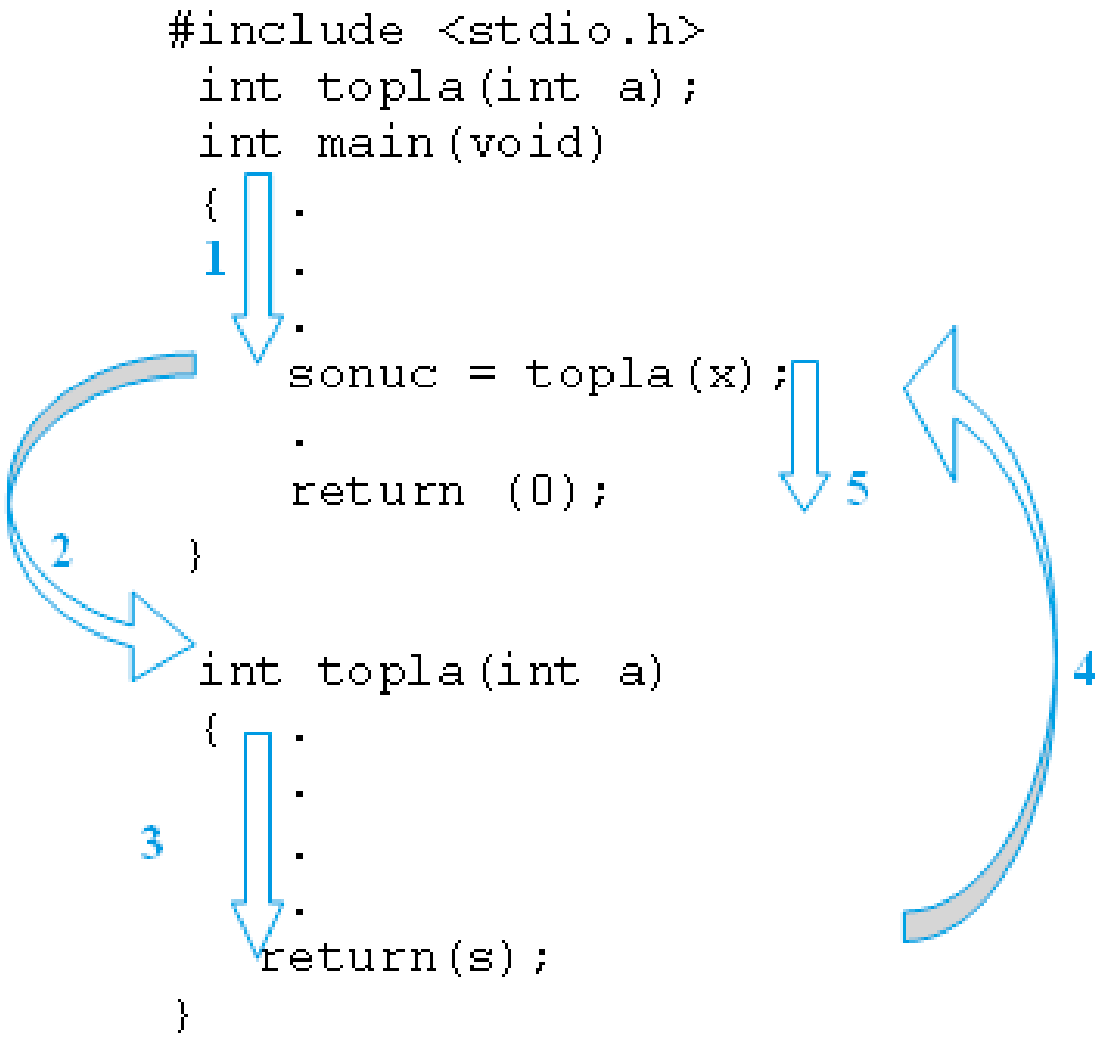
```
#include <stdio.h>
int main(void)
{    int x,i,sonuc;
     sonuc =0;
     printf("Bir tamsayi giriniz:");
     scanf("%d",&x);
     for(i=1;i<=x;++i)
          sonuc = sonuc+i;
     printf("Toplam=%d", sonuc);
     return (0);
}
```

→ Fonksiyon başlığı

→ Fonksiyon gövdesi

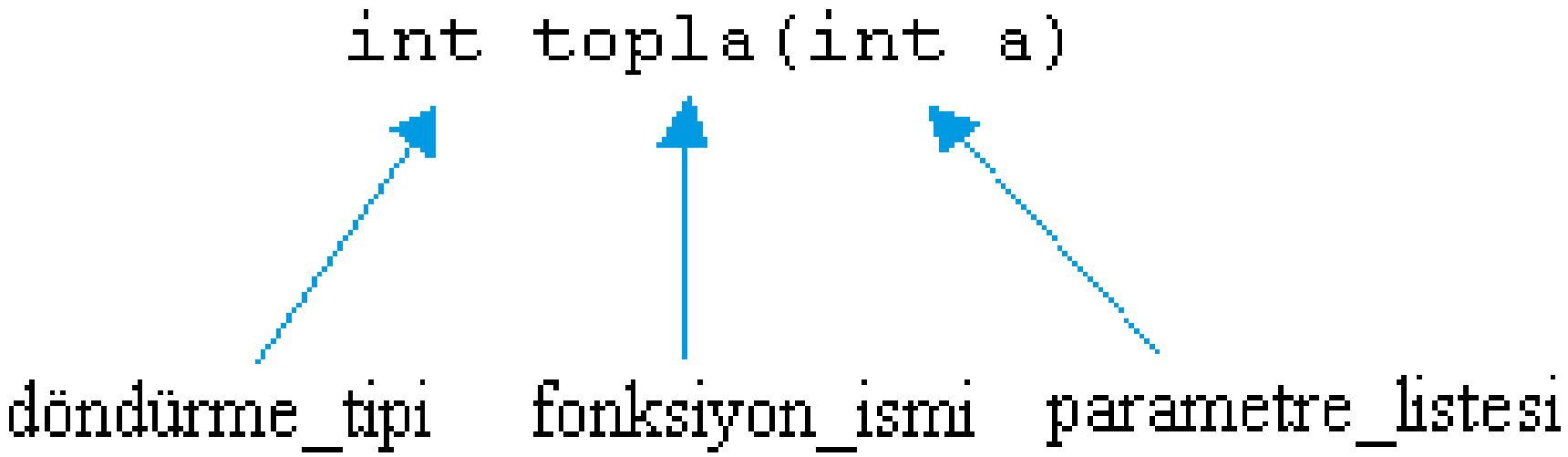
Fonksiyon Tanımı

Örnek: İki fonksiyondan oluşan bir program



Fonksiyon Başlığı

döndürme_tipi fonksiyon_ismi (parametre_ listesi)



Fonksiyon Prototipi

Fonksiyon başlığına benzer, cümle sonunda noktalı virgül vardır. Fonksiyon prototipi program başında yer almalıdır. Böylece, derleyici, fonksiyon prototipini gördüğünde fonksiyon ismini tanır.

```
int topla(int a);
```
veya
```
int topla(int);
```

Değer Döndürmeyen Fonksiyonlar

Parametresiz Fonksiyonlar

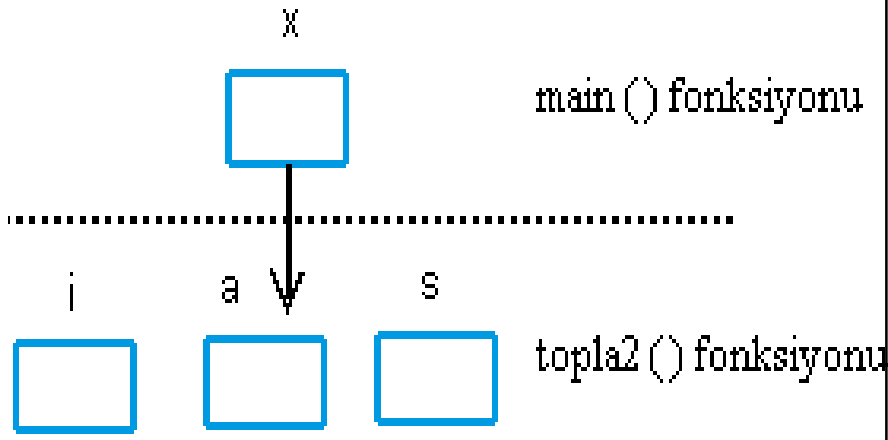
Parametreler, fonksiyonların çağırıldıkları yerden veri almasını sağlayan değişkenlerdir. Parametresiz, değer döndürmeyen fonksiyonlar çağırıldıkları yerden bir veri almayan ve herhangi bir veri geri döndürmeyen fonksiyonlardır.

```
1  #include <stdio.h>
2  void topla1(void);      → Fonksiyon prototipi
3  int main(void)
4  {    topla1();          → Fonksiyon çağırma
5       return (0);
6  }
7
8  void topla1(void)       → Fonksiyon tanımı
9  {    int i,s,x;
10      s=0;
11      printf("Bir tamsayi giriniz:");
12      scanf("%d",&x);
13      for(i=1;i<=x;++i)
14          s=s+i;
15      printf("Toplam = %d", s);
16 }
```

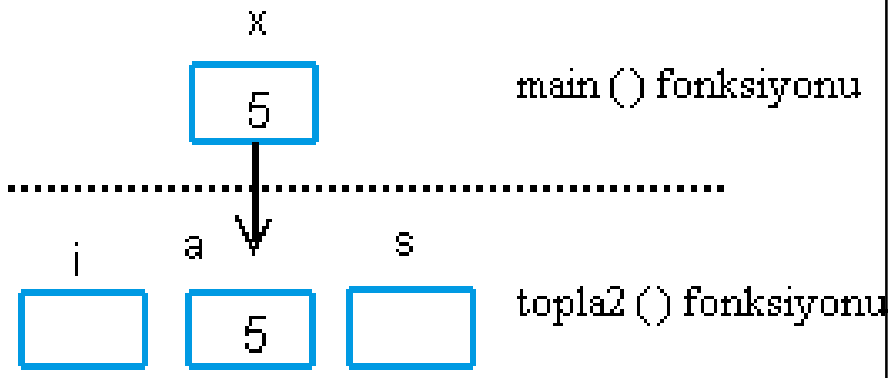
Değer Döndürmeyen Fonksiyonlar

Parametreli Fonksiyonlar çağırıldıkları yerden veri alan ama herhangi bir veri geri döndürmeyen tip fonksiyonlardır.

```
 1  #include <stdio.h>
 2  void topla2(int a);      → Fonksiyon prototipi
 3  int main(void)
 4  {    int x;
 5       printf("Bir tamsayi giriniz:");
 6       scanf("%d",&x);
 7       topla2(x);          → Fonksiyon çağırma
 8       return (0);
 9  }
10
```




```
11  void topla2(int a)       → Fonksiyon tanımı
12  {    int i,s;
13       s=0;
14       for(i=1;i<=a;++i)
15            s=s+i;
16       printf("Toplam = %d", s);
17  }
```



Değer Döndüren Fonksiyonlar

Parametresiz Fonksiyonlar çağırıldıkları yere bir değer döndüren ve çağırıldıkları yerden bir veri almayan fonksiyonlardır.

```
1   #include <stdio.h>
2   int  topla3(void);          → Fonksiyon prototipi
3   int main(void)
4   {   int sonuc;
5       sonuc =topla3();        → Fonksiyon çağırma
6       printf("Toplam = %d", sonuc);
7       return (0);
8   }
9
10  int topla3(void)            → Fonksiyon tanımı
11  {   int i,s,x;
12      printf("Bir tamsayi giriniz:");
13      scanf("%d",&x);
14      s=0;
15      for(i=1;i<=x;++i)
16          s=s+i;
17      return(s);
18  }
```

Değer Döndüren Fonksiyonlar

Parametreli Fonksiyonlar hem çağırıldığı yerden değer alan, hem de çağırıldığı yere değer döndüren fonksiyonlardır.

```
#include <stdio.h>
int topla(int a);                    → Fonksiyon prototipi
int main(void)
{    int x,sonuc;
     printf("Bir tamsayi giriniz:");
     scanf("%d",&x);
     sonuc = topla(x);               → Fonksiyon çağırma
     printf("Toplam = %d", sonuc);
     return (0);
}


int topla(int a)                     → Fonksiyon tanımı
{    int i,s;
     s=0;
     for(i=1;i<=a;++i)
          s=s+i;
     return(s);
}
```

Çok Parametreli Fonksiyonlar

Fonksiyon Başlığı

döndürme_tipi fonksiyon_ismi (formal_parametre_listesi)

Fonksiyonu çağırma cümlesi

fonksiyon_ismi (gerçek_ parametre_listesi);

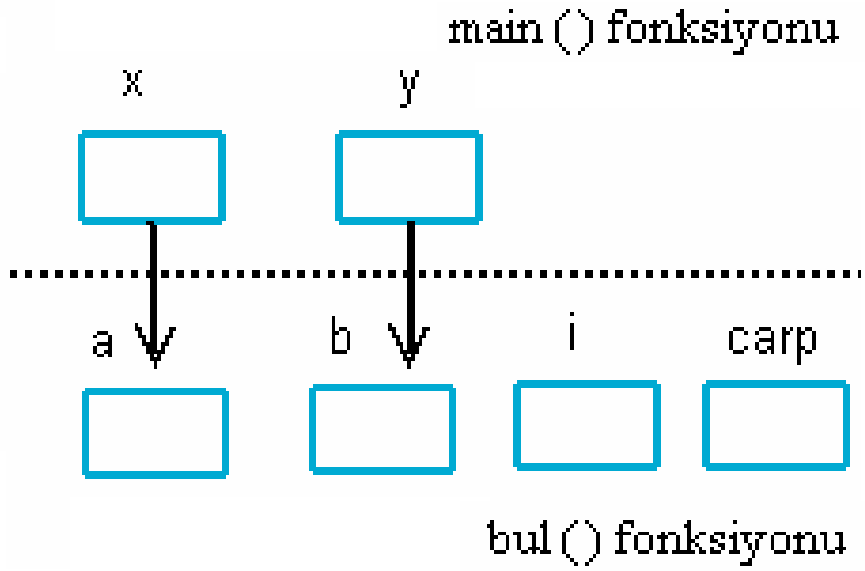
Çok Parametreli Fonksiyonlar

```
 1  #include <stdio.h>
 2  void bul(int a, int b);
 3  int main(void)
 4  {    int x,y;
 5       printf("Iki pozitif tamsayi giriniz:");
 6       scanf("%d %d",&x,&y);
 7       bul(x,y);
 8       return (0);
 9  }
10
11  void bul(int a,int b)
12  {    int i, carp;
13       carp=1;
14       for(i=1;i<=b;++i)
15         carp=carp*a;
16       printf("Sonuc = %d", carp);
17  }
```

→ Fonksiyon prototipi

→ Fonksiyon çağırma

→ Fonksiyon tanımı



Değişken Kapsamı

Değişken kapsamı, bir değişkenin tanımının programın hangi bölümünde geçerli olduğunu veya tanındığını gösterir.

Yerel kapsam, değişkenin tanımlandığı fonksiyona ait olduğunu ve değişkene sadece tanımlandığı fonksiyon içinden erişilebilindiğini gösterir. Bu tip değişkenlerin tanımı fonksiyon içinde yapılır ve bu değişkenler fonksiyonun bitiminde bellekten silinirler.

Genel kapsamlı bir değişken, fonksiyonların dışında tanımlanır ve tanımlamanın yapılmasından sonra yazılmış tüm fonksiyonlar tarafından tanınır ve kullanılır.

Değişken Kapsamı

```
 1  #include <stdio.h>
 2  void f(int k);
 3  int a=3;
 4  int main(void)
 5  { int x,y;
 6    x=10;
 7    y=5;
 8    printf("%3d%3d%3d\n",x,y,a);
 9    f(x);
10    a=a+1;
11    printf("%3d%3d%3d\n",x,y,a);
12    return (0);
13  }
14  void f(int k)
15  { int s;
16    s=k+1;
17    k=k*3;
18    a=a+1;
19    printf("%3d%3d%3d\n",k,s,a);
20  }
```

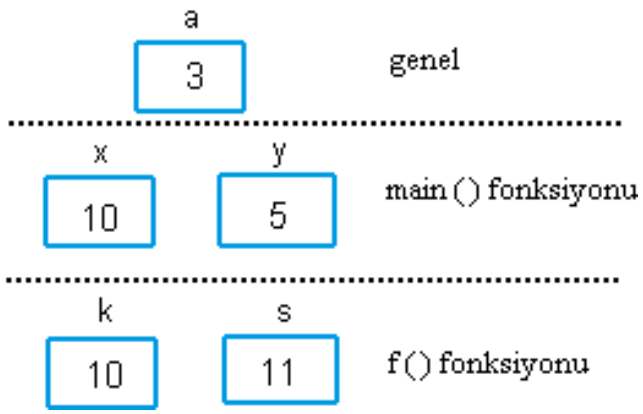
Genel değişken tanımlaması: a değişkeni main() ve f() tarafından kullanılabilir.

Yerel değişken tanımlaması: x ve y sadece main() tarafından kullanılabilir.

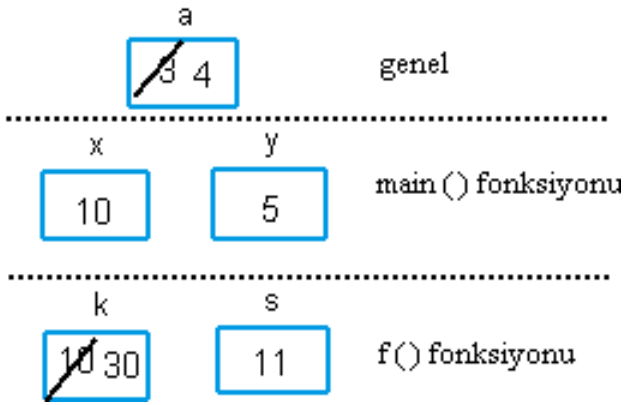
Yerel değişken tanımlaması: k ve s sadece f() tarafından kullanılabilir.

Değişken Kapsamı

f() fonksiyonu çağırıldığında



f() fonksiyonu içinde



Çıktı

```
10   5   3
30  11  4
10   5   5
```

Çok Fonksiyonlu Programlar

Bir program birden fazla fonksiyondan oluşursa, main() fonksiyonu, istenen fonksiyonu çağırabilir veya bir fonksiyon başka bir fonksiyonu çağırabilir.

```
#include <stdio.h>
int faktoriyel (int  k);
int kombinasyon (int  n, int r);
int main(void)
{       int k,s,cevap;
        printf("Iki tamsayi giriniz:");
        scanf("%d%d",&k,&s);
        cevap=kombinasyon(k,s);
        printf("Kombinasyon=%3d ",cevap);
        return(0);
}
int faktoriyel (int  k) /*Faktoriyel hesabi*/
{       int i, s=1;
        for(i=1;i<=k;++i)
                s*=i;
        return (s);
}
int kombinasyon (int  n, int r) /*Kombinasyon hesabi*/
{       int s;
        s=faktoriyel(n)/(faktoriyel(r)*faktoriyel(n-r));
        return (s);
}
```