

PROGRAMLAMA TEMELLERİ

BÖLÜM 8

ÖĞR. GÖR. HAKAN CAN ALTUNAY

C DİLİNDE FONKSİYONLAR

C dilinde programlar bloklar halinde yazılır. Büyük programlar ise parça parça fakat birbirleriyle ilişkili fonksiyonlar halinde yazılırlar. Bunun avantajı daha sonra programı üzerinde kolaylıkla değişiklik yapılabilecek olmasıdır. C dilinde fonksiyonlar bir veya birden fazla işlemden oluşan alt programlar olarak da düşünülebilir.

C dilindeki ana program olan main ()'de aslında bir fonksiyondur. Fonksiyonların genel yapısı aşağıda gösterildiği gibidir.

dönüş_tipi / void fonksiyonadi ([varsa] parametreler)

```
{  
  
    [varsa] yerel tanımlamalar  
  
    işlem1;  
  
    işlem2;  
  
    [varsa] return değer;  
  
}
```

Dönüş tipi: Eğer fonksiyon bir değer geri döndürecek ise değerın tipi belirtilir ve bu değer return deyimi ile döndürölür. Döndürölün değeri fonksiyon dönüş tipiyle aynı olmalıdır. Değeri döndürmeyecek ise dönüş tipi yerine void yazılır ve return ifadesiyle herhangi bir değeri döndürölmez.

Fonksiyon adı: Fonksiyon çağrılırken kullanılacak belirleyici isimdir. Değışkenler tanımlanırken sylenen kurallar, fonksiyon tanımlanırken de geçerlidir.

Yerel Tanımlamalar: Fonksiyona özğü değışken veya sabitlerin tanımlandığı yerdir.

Fonksiyon Prototipi: Tanımlanan bir fonksiyonun programın başında prototipinin yazılması gerekir. Ön bildirim ile fonksiyona çağrıda bulunmadan önce derleyicinin böyle bir fonksiyondan haberi olmalıdır.

```
int kup(int x);
```

Yukarıdaki komut satırında kup fonksiyonu int bir x değeri alır ve ine int bir değer döndürür.

Fonksiyonlar değer döndüren ve değer döndürmeyen (void) olmak üzere ikiye ayrılır.

1- Değer Döndürmeyen Fonksiyonlar:

Bu fonksiyonlar alt program gibi kullanılarak ana fonksiyon veya diğer fonksiyonlar tarafından çağrılır. Fonksiyon içindeki işlemler icra edilir. Fakat çağrıda bulunan fonksiyona herhangi bir değer döndürmezler.

```
#include <stdio.h>
void yaz(); /* fonksiyon ön bildirimi */
void main()
{   yaz();
}

void yaz()
{   printf("C Programlama Notları\n");
}
```

Program çıktısı:
C Programlama Notları

Yukarıdaki program çalıştırıldığında yaz () fonksiyon ismi kullanılarak fonksiyon çağrılır, yaz () fonksiyonu ana fonksiyon ile aynı ortamda yazılmış ve derlenmiştir. yaz() fonksiyonu kullanılarak ekrana “C Programlama Notları” yazdırılır. Bu işlem tamamlandıktan sonra ana programa herhangi bir değer aktarılması söz konusu değildir.

Tanımlanan fonksiyonlara, çağrıda bulunmak, kullanılmak üzere parametreler yardımıyla değer aktarmaktır. Bu değerler fonksiyon tarafından kullanılır ve verilen görevi yerine getirir. Fakat her zaman fonksiyon tarafından üretilen değer çağrıda bulunan fonksiyona aktarılmayabilir. Çağrıda bulunan fonksiyona aktarılan değerler, fonksiyon tanımlanırken fonksiyon adının yanında parantez içerisinde parametre olarak listelenir. Aşağıda örnek programda carp () fonksiyonuna ana fonksiyondan değer aktarılmaktadır.

```

#include <stdio.h>
void carp(int x, int y); /* fonksiyon ön bildirimi */
void main()
{
    carp(10,20);
    carp(5,6);
    carp(8,9);
}

void carp(int x, int y) /* fonksiyon tanımlaması */
{
    printf ("%2d * %2d = %3d\n", x, y, x*y);
}

```

Program çıktısı:

```

10 * 20 = 200
 5 *  6 =  30
 8 *  9 =  72

```

Yukarıdaki program çalıştırıldığında carp() fonksiyonunun ilk defa kullanıldığı satırda 10 ve 20 sayıları parametre olarak verilen int tipi x ve y değişkenlerine gönderilir, x değişkenine 10, y değişkenine 20 değeri aktarılır. Fonksiyonun ikinci ve üçüncü çağrılmalarında sırasıyla x ve y değişkenlerine 5 ve 6 ile 8 ve 9 değerleri aktarılır. Ve bunların çarpımları ekrana yazdırılır.

C programlama dilinde değişkenler tanımlandıkları yere göre global lokal olmak üzere ikiye ayrılır. Programın başında tanımlanıp tüm fonksiyonlar tarafından değerine erişilebilen değişkenlere global değişken, herhangi bir fonksiyon içinde tanımlanıp sadece tanımlandığı fonksiyonda geçerli olan ve sadece bu fonksiyon içerisinde değerine erişilebilen değişkenlere lokal değişken denir.

```

#include<...>
.... fonk(...);
int a; /*tüm fonksiyonlardan değerine erişilebilir,
        değeri değiştirilebilir. (global) */

void main()
{
    int b; /* sadece main fonksiyonu içerisinde
            erişilebilir (local ) */
    ...
}

.... fonk(...)
{
    int b; /* sadece fonk fonksiyonu içerisinde
            erişilebilir. main fonksiyonundaki
            b değişkenden bağımsızdır (local ) */
    int c; /* sadece fonk fonksiyonu içerisinde
            erişilebilir (local ) */
    ...
}

```

Fonksiyonun içinde tanımlanan bir dğeişkenin sadece o fonksiyon içinde geçerli olduğu bilinmektedir. Hatta aynı isimli değişken hem ana programda hem de

fonksiyon içinde tanımlanabilir ve bu değişkenlerin isimlerinin aynı olmasına rağmen farklı değerler taşıyabilir. Aşağıdaki programda x değişkeni hem ana programda hem de fonksiyon içinde tanımlanmış olmasına rağmen iki ayrı değişken olarak kullanılmaktadır.

```
#include <stdio.h>
void fonk();

void main() /* ana program */
{
    int x;
    x=10;
    printf("ana programdaki x degeri: %d\n", x);
    fonk();
    printf("ana programdaki x degeri: %d\n",x);
}

void fonk() /* fonksiyon */
{
    int x;
    x=100;
    printf("fonksiyondaki x degeri:  %d\n", x);
    return;
}
```

Program çıktısı:

```
ana programdaki x degeri: 10
fonksiyondaki x degeri:  100
ana programdaki x degeri: 10
```

2- Değer Döndüren Fonksiyonlar:

Bu fonksiyonlarda, fonksiyon tarafından üretilen değer, return ifadesi ile çağrıda bulunan fonksiyona döndürülmektedir ve bu değer orada kullanılmaktadır.

```
#include <stdio.h>
int carp(int x, int y); /* fonksiyon ön bildirimi */
void main() /* ana program */
{
    int a, b, c;
    printf ("Birinci sayiyi giriniz:");
    scanf("%d", &a);
    printf("Ikinci sayiyi giriniz:");
    scanf("%d", &b);
    c=carp(a, b);
    printf("Carpim: %d\n", c);
    /* veya printf("Carpim: %d\n", carp(a, b)); */
}

int carp(int x, int y) /* fonksiyon */
{
    int sonuc;
    sonuc= x * y;
    return sonuc; /* veya return x*y; */
}
```

Program çıktısı:

```
Birinci sayiyi giriniz: 6
Ikinci sayiyi giriniz: 7
Carpim: 42
```

Yukarıdaki programda ana programdan girilen 6 ve 7 değerleri x ve y değişkenlerine gönderilmektedir. Bu sayılar fonksiyonda çarpılarak elde edilen 42 değeri c değişkeninde tutulmaktadır. Return c ile ise lokal c değişkeninde tutulan değer ana programa aktarılmakta ve aktarılan bu değer ana programda tanımlanmış sonuç değişkenine aktarılarak program tarafından görüntülenmektedir.

KAYNAKLAR:

- 1- ALGORİTMA GELİŞTİRME VE PROGRAMLAMAYA GİRİŞ, FAHRİ VATANSEVER, SEÇKİN YAYINEVİ**
- 2- VERİ YAPILARI VE ALGORİTMALAR, RİFAT ÇÖLKESEN, PAPATYA BİLİM YAYINEVİ**
- 3- ALGORİTMA VE PROGRAMLAMA MANTIĞI, BURAK TUNGUT, KODLAB**