

İskenderun Teknik Üniversitesi



Fakülte: Mühendislik ve Doğa Bilimleri *Fakültesi*

Bölüm: *Bilgisayar Mühendisliği Bölümü*

Ders: Algoritmalar ve Programlama

Dönem: 2021 – 2022 (Güz)

Öğretim üyesi: *Dr. Öğr. Üyesi Mehmet SARIGÜL*

8.

Hafta

- Gerçek dünyadaki sorunları çözen çoğu bilgisayar programı, ilk birkaç bölümde sunulan programlardan çok daha büyüktür.
- Deneyimler, büyük bir programı geliştirmenin ve sürdürmenin en iyi yolunun, onu her biri orijinal programdan daha yönetilebilir olan daha küçük parçalardan inşa etmek olduğunu göstermiştir.
- Bu tekniğe böl ve fethet denir.

Giriş

- Programları modüler hale getirmek için fonksiyonlar (Prosedürler, İşlevler) kullanılır
- C programları tipik olarak, yazdığınız yeni işlevler ile C standart kitaplığında bulunan önceden paketlenmiş işlevler birleştirilerek yazılır.
- C standart kitaplığı, yaygın matematiksel hesaplamaları, dize işlemlerini, karakter işlemlerini, giriş / çıkış ve diğer birçok yararlı işlemi gerçekleştirmek için zengin bir işlev koleksiyonu sağlar.

Giriş

- Zaten kendi fonksiyonumuzu yazdık ve kütüphane fonksiyonlarını kullandık:
 - main, her C programında bulunması gereken bir işlevdir.
 - cout, cin zaten programlarımızda kullandığımız kütüphane işlevleridir.
- Bir programın birçok noktasında kullanılabilecek görevleri tanımlamak için kendi fonksiyonlarımızı yazabiliriz.
- Bunlar bazen programcı tanımlı fonksiyonlar olarak adlandırılır.

Fonksiyonlar

- Fonksiyonlarla iki şey yapmamız gerekiyor:
 - Fonksiyonu oluşturun.
 - Fonksiyonu çağırın!
- Fonksiyonlar, fonksiyon adını belirten ve çağrılan fonksiyonun belirlenen görevi yerine getirmesi için gereken bilgileri (bağımsız değişkenler olarak) sağlayan bir işlev çağrısı tarafından çağrılır.

Fonksiyonlar

donus_tipi **fonksiyon_adi** (**parametre-tanımlamaları**)

```
{  
    değişken-tanımları;  
    komut-ifadeleri;  
}
```

Fonksiyon yazımı

- dönüş_türü - işlevin türünü belirtir ve işlev tarafından döndürülen değerin türüne karşılık gelir
 - void - işlevin hiçbir şey döndürmediğini belirtir.
 - belirtilmezse, int türü döner.
- fonksiyon_adı - tanımlanmakta olan işlevin adı (herhangi bir geçerli tanımlayıcı)
- parametre bildirimleri - işlevin parametrelerinin (biçimsel parametreler) türlerini ve adlarını virgülle ayırarak belirtin.

Fonksiyon tanımlama



```
int cube ( int num )  
{  
    int result;  
    result = num * num * num;  
    return result;  
}
```

- `n = cube(5);`



```
void message(void)
{
    printf("A message for you:");
    printf("Have a nice day!\n");
}
```

- message();

- İşlev tanımlarında tanımlanan tüm değişkenler yerel değişkenlerdir - bunlara yalnızca tanımlandıkları işlevde erişilebilir.
- Çoğu işlevin, işlevler arasında bilgi iletişimi için araçlar sağlayan bir parametre listesi vardır.
- Bir fonksiyonun parametreleri aynı zamanda o fonksiyonun yerel değişkenleridir.

Fonksiyonlar

- Bir return ifadesi yürütüldüğünde, işlevin yürütülmesi sonlandırılır ve program kontrolü hemen çağıran ortama geri gönderilir.
- Bir ifade return anahtar sözcüğünden sonra gelirse, ifadenin değeri çağıran ortama da döndürülür.
- Bir dönüş ifadesi aşağıdaki iki formdan biri olabilir:
 - return;
 - return değişken;

return ifadesi

- *return;*
- *return 1.5;*
- *return result;*
- *return a+b*c;*
- *return x < y ? x : y;*

Örnekler



```
int IsLeapYear(int year)
{
    return ( ((year % 4 == 0) && (year % 100 != 0))
            || (year % 400 == 0) );
}
```

- Çağrı bu şekilde olabilir;
- *if (IsLeapYear(2005))*
- *printf("29 days in February.\n");*
- *else*
- *printf("28 days in February.\n");*

- İki sayı girilince minimumunu ekrana yazdıran programı yazınız;

```
int min(int a, int b)
{
    if (a < b)
        return a;
    else
        return b;
}
```


- Bir işlevin sıfır veya daha fazla parametresi olabilir.
- Bildirim başlığında:
 - `int f (int x, double y, char c);`
- İşlev çağrısında:
 - `değer = f (yaş, puan, harf);`

Parametreler

- Gerçek ve biçimsel parametre listelerindeki parametre sayısı tutarlı olmalıdır
- Parametre ilişkilendirmesi konumsaldır: ilk gerçek parametre ilk biçimsel parametre ile eşleşir, ikincisi ikinciyle eşleşir vb.
- Gerçek parametreler ve resmi parametreler, uyumlu veri türlerinde olmalıdır
- Gerçek parametreler bir değişken, sabit olabilir, karşılık gelen biçimsel parametrenin türüyle eşleşen herhangi bir ifade olabilir.

Parametreler

- **Output:**

- 5
- 0
- 5
- 15

Örnek

```
#include <stdio.h>
int compute_sum (int n)
{
    int sum;
    sum = 0;
    for (n; n > 0; n--)
        sum += n;
    printf("%d\n", n);
    return sum;
}

int main (void)
{
    int n, sum;
    n = 5;
    printf("%d\n", n);
    sum=compute_sum(n);
    printf("%d\n",n);
    printf("%d\n", sum);
    return 0;
}
```

Örnek

```
/* Finding the maximum of three integers */
#include <stdio.h>
int maximum( int x, int y, int z ); //function prototype

int main()
{
    int a, b, c;
    printf( "Enter three integers: " );
    scanf( "%d%d%d", &a, &b, &c );
    printf( "Maximum is: %d\n", maximum( a, b, c ) );
    return 0;
}

//function definition
int maximum( int x, int y, int z )
{
    int max = x;
    if ( y > max )
        max = y;
    if ( z > max )
        max=z;
    return max;
}
```




```


#include <stdio.h>
int max(int,int);
int min(int,int);

int main(void)
{
    ...
    min(x,y);
    max(x,y);
    ...
}

int max (int a, int b)
{
    ...
}

int min (int a, int b)
{
    ...
}

```



```

#include <stdio.h>

int max (int a, int b)
{
    ...
}

int min (int a, int b)
{
    ...
}

int main(void)
{
    ...
    min(x,y);
    max(x,y);
    ...
}

```



Ders Sonu

