

ALGORİTMA VE PROGRAMLAMA (Bölüm 3)

C PROGRAMLAMA DİLİ

- C Nasıl Bir Programlama Dilidir?
 - **C** orta düzey bir programlama dili olarak son derece **esnek kullanımlı bir dildir.**
 - **C**, bir **Assembly** dili derleyicisinin sağladığı **esneklik ve gücü sağlarken** üst düzey bir programlama dilinin sağladığı **kolay anlaşılabilirlik özelliğini** de sunmaktadır

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- **C Dilinin Temel Özellikleri :**
 - **C, oldukça fazla sayıda ifade ve denetim komutları bulunduran bir programlama dilidir.**
 - **Ancak, dikkatli yazılmalıdır.**
 - **Zira hatalı veya eksik bir işaret programımızı doğru çalıştırmayacaktır.**

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

■ Program Yazma, Derleme ve Çalıştırma:

- Önemli olan, önce **basit bir programı yazıp derleyebilmek, çalıştırabilmek** ve elde edilen **sonuçları** görebilmektir.
- **Yazma, Derleme ve Çalıştırma** kullandığımız ortama bağlı olmakla birlikte genellikle kolay bir şekilde yapılabilir.

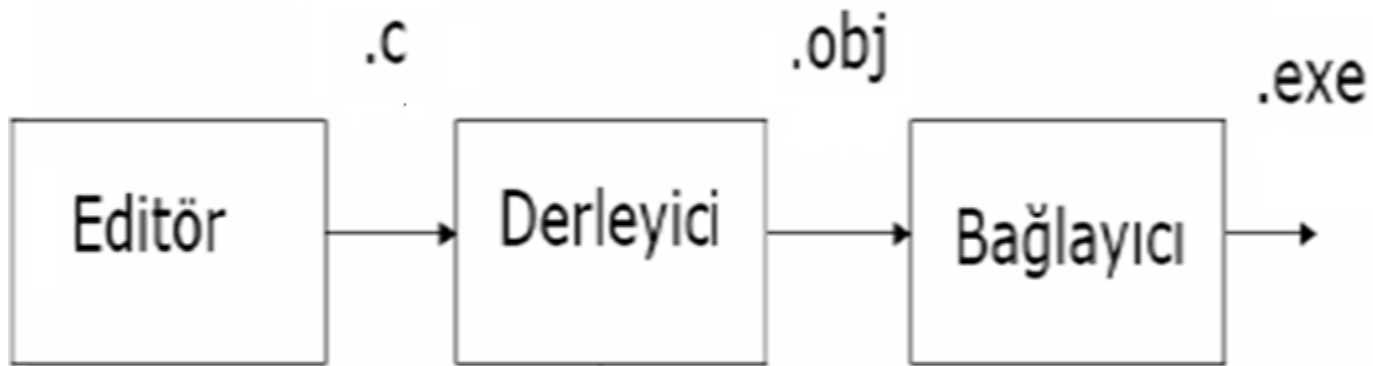
C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- C dilinde yazılan programın **uzantısı (.c)** olur.
- C dilindeki program **(derleyici, bağlayıcı ve Kütüphane fonksiyonları ile) önce (.obj), sonra (.exe)** uzantılı olur...
- Böylece **(.exe)** uzantısı dolayısıyla PC 'de **yürütülebilir** hale gelir.

C PROGRAMLAMA DİLİ (*Devam*)

Başlarken...



C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- C 'de bulunan pek çok **fonksiyonun** kullanılabilmesi için bazı **kütüphane başlık dosyalarının (include) programa eklenmesi** gerekir.
- Karışıklığı önlemek adına programlarımızda gösterilmeyecek olan bu dosyalardan en **temel ve önemli olanı:**

#include <stdio.h> şeklindedir ve...

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- Bu başlık dosyasından başka **ileri matematik** işlemlerde kullanılan...

#include <math.h>

- Yine rastgele **sayı üretmek, sıralama yapmak vb.** işlemlerde...

#include <stdlib.h>

gibi farklı başlık dosyaları ekleme ifadeleri mevcuttur...

- İhtiyaca göre bu ifadelerin **main()** öncesinde programın başına eklenmesi gerekir.

C PROGRAMLAMA DİLİ (*Devam*)

Başlarken...

- **IDE (Tümleşik Geliştirme Ortamı):**
 - Tümleşik Geliştirme Ortamları, **Derleyicileri–Bağlayıcıları tek bir ortam içinden kullanabilmeyi sağlarken...**
 - Ayrıca **Hata ayıklama** (debug), **Gözlem penceresi** (watch) gibi bileşenlere sahiptirler.

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- **IDE'lerde yazım işlemini kolaylaştıracak vurgulamalar, uyarılar ve yardım menüleri mevcuttur.**
- **Derleyici ve Bağlayıcı parametreleri menülerden ayarlanabilir.**
- **Her yazılım dilinin kendi IDE 'si mevcuttur...**

C PROGRAMLAMA DİLİ (Devam)

Başlarken...

- **İlk Program**: Programlamayı öğrenmeye başlarken sıkça kullanılan bir ifade olan “**Merhaba C**” yi ekrana yazdıran bir **mini örnek programla başlayalım**:

```
main() {  
    printf(“Merhaba C\n”) ;  
}
```

Not: Daha evvel de belirtildiği gibi yukarıdaki **printf()** fonksiyonunun çalışabilmesi için **main()** den önce **#include <stdio.h>** yazarak kütüphane başlık dosyasının eklenmesi gerekir.

*Dersimizdeki örneklerde bu dosya eklenmeyecektir ancak bilgisayarda derleme yapmadan önce programda **kullanılan fonksiyonlara** göre bu ve benzer dosya/dosyaların ilave edilmesi şarttır.*

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- Bu programın Ekran Çıktısı;

Merhaba C

şeklinde olur.

Not:

- **main** İngilizcede **ana, asıl** anlamındadır.
- **printf()** ekrana yazdırmak için kullanılan bir fonksiyondur.

C PROGRAMLAMA DİLİ (*Devam*)

Başlarken...

- Her komuttan (fonksiyondan/deyimden) sonra **;** **işareti konulması şarttır.**
- **\n** Merhaba C ifadesinden sonra satır atlatır.
- **{ ve }** küme parantezleridir. main() fonksiyonu fonksiyonları/deyimleri bunların içine yazılır.

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- Bir **C** programı boyutu ne olursa olsun **bir ya da daha fazla fonksiyon denilen bloktan** oluşur.
- **C** Dilinde olmazsa olmaz fonksiyon (blok) **main()** fonksiyonudur (bloğudur *).
- Program **her zaman bu main() adlı fonksiyondan çalışmaya başlar.**

Not: Bundan sonra blok yerine “fonksiyon” ifadesi kullanılacaktır.

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- Genelde fonksiyonun yanındaki **parantez içinde parametreler (argümanlar)** yer alır. *Örn:*

```
printf("Merhaba C\n");
```

ifadesindeki **Merhaba C\n** gibi...

Not: *main()* fonksiyonunun herhangi bir parametresi olmadığından **parantezi boştur**. Parametre yoksa parantez içine **void** ifadesi de yazılabilir.

C PROGRAMLAMA DİLİ *(Devam)*

Başlarken...

- **Fonksiyonlar arasında haberleşme ve (varsa) değerlerin transferi parametreler (argümanlar) maharetiyle** gerçekleşir.
- **Fonksiyonlar adı yazılan yerde** çağrılır(çalışır).
- (Varsa) elde edilen sonuçlar **kendisini çağırın fonksiyona** gönderilir.

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu

❖ printf() FONKSİYONU

```
printf("Merhaba C \n");
```

satırı (deyimi) adı **printf** olan bir fonksiyon çağırır.

Not : ***printf()*** ekrana bir **bilginin yazılmasını** sağlayan **bir kütüphane fonksiyonudur ve standart C arşivinin (kütüphanesinin) bir parçasıdır.**

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu

- **printf()** fonksiyonu asla otomatik olarak **yeni bir satıra geçiş sağlamaz.**
- Aşağıdaki program **önceki örnekteki ile aynı çıktıyı** verecektir :

```
main() {  
    printf("Merhaba ");  
    printf("C ");  
    printf("\n");  
}
```

C PROGRAMLAMA DİLİ *(Devam)*

Bazı Kontrol Karakterleri

- Benzer şekilde...
 - **\n** satır atlattığı gibi...
 - **\t** tab için,
 - **\b** geriye doğru silme (kürsörü bir karakter öne alma) işlemi için (Backspace),
 - **\"** çift tırnağın kendisini yazdırmak için kullanılır.
- Bunlar görünmeyen **kontrol karakterlerdir**.
- Ekrandaki **çıktıyı kontrol** etmektedirler.

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu...

- **Mesaj ile birlikte bir değişkenin değerini** ekrana yazdırmak için yine **printf()** fonksiyonu kullanılabilir.
- İlk örnek **“Merhaba C”** de olduğu gibi sadece **mesaj ve değişkenin biçimi çift tırnak içine** alınarak belirtilir.

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu...

- **printf()** fonksiyonu ile **hem mesaj hem de değişken değeri birlikte yazılabilir.**
- **Örnek:** Ekranda **sonuc** adlı desimal bir değişken yazdırılacaktır. Bunun için önce **sonuc=** ifadesi, bu ifadenin hemen yanına bu değişkenin değeri yazılacaktır. Değişken değerinden sonra **dir.** eklenecektir.
Gerekli **printf()** deyimini (komutunu) yazınız.

C PROGRAMLAMA DİLİ *(Devam)* **printf()** fonksiyonu...

Cevap:

```
printf("sonuc= %d dir.",sonuc);
```

*Not: Yukarıdaki **printf()** fonksiyonu ile **sonuc** adlı değişkenin değeri **sonuc=** ifadesinin yanına (%d nin yerine) yazılmakta sonra da **dir.** eklenmektedir.*

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu...

- Burada **%d** , değerin **desimal olarak** yazılacağını belirtir.
- Ekranda **sonuc** değişkeninin değeri komuttaki **%d** nin yerine yazılır.
- Bu durumda **Ekran Çıktısı**, (*sonuc değeri 25 olsun*)

sonuc=25 dir.

şeklinde olur.

C PROGRAMLAMA DİLİ *(Devam)* **printf()** fonksiyonu...

- **Örnek:** Daha önce hesaplanmış **m** değişkeni **Sayi=** ve **n** değişkeni **Karekoku=** ifadelerinden sonra ekrana yazdırılmak istenmektedir. **m** değişkeni **desimal** (tam sayı), **n** değişkeni **ondalıklı** (virgüllü) sayı olacaktır. Tek bir **printf()** deyimini ve Ekran Çıktısını yazınız.

Not: *Ondalıklı sayı için %f kontrol karakteri kullanılır.*

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu...

Cevap:

```
printf("Sayi= %d Karekoku=%f ",m,n);
```

- **Ekran Çıktısı :**

(m=5 ve n= 2.236076 değişken değerleri girildiği durumda)

Sayi= 5 Karekoku= 2.236076

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu...

- Yukarıda görüldüğü gibi **%d** , **%f** vb. gibi yazdırma kontrol karakterleri dışındaki **çift tırnak içindeki karakterler çıkışta aynen** yazılır.
- Parantezde kaç değişken varsa o kadar kontrol karakteri olmalıdır. (Örnekte: **m** ve **n** için **%d** ve **%f**)

Not : *Eğer kesirli olan (n) değişkeni için kontrol karakteri olarak %f yazılmayıp %d şeklinde yazılsaydı **2.236076** sayısı yerine **çıkıtı**da sadece **2** (tam sayı kısmı) olarak görülecekti.*

C PROGRAMLAMA DİLİ *(Devam)*

printf() fonksiyonu...

- **printf()** komutu içinde yaygın olarak kullanılan...
bazı **giriş veya yazdırma kontrol karakterleri** şöyledir:
- **%c** : tek karakter [örn: M]
- **%d** : desimal (tam sayı) [örn: -326]
- **%u** : işaretsiz desimal [örn: 476] (*Sadece pozitif sayılar*)
- **%f** : ondalık kayan nokta [örn: 314.2857]
- **%e** : bilimsel (üslü) gösterim [örn: 3.142857e+002]
- **%x** : heksadesimal [örn: 3F]
- **%s** : karakter katarı [örn: ADRES]

C PROGRAMLAMA DİLİ *(Devam)*

scanf() fonksiyonu

❖ **scanf() FONKSİYONU:**

- **Klavyeden değer okumak için**
scanf() adlı yine bir **standart kütüphane fonksiyonu** kullanılabilir.
- Bu fonksiyonda da benzer **kontrol karakterleri** vardır ve **okunacak değerlerin tipini** belirler.
- **printf()**'e **benzer** şekilde **%d** tam sayı, **%f** kesirli (virgüllü) sayı okunacağını ifade eder.

C PROGRAMLAMA DİLİ *(Devam)*

scanf() fonksiyonu

- Mesela iki **scanf()** fonksiyonu ile önce **a** değişkenine bir **tam** , sonra **b** değişkenine bir **kesirli** sayı okunacaksa;

```
scanf (“%d”, &a);
```

```
scanf (“%f”, &b); yazılabilir
```

Not: Burada **a** veya **b** adlı değişkenlerin hafızadaki değerleri (içeriği) anlamında başlarına **&** karakteri gelmektedir.

C PROGRAMLAMA DİLİ (Devam)

scanf() fonksiyonu

- Eğer **X** ve **Y** adlı iki desimal değişkeni aynı scanf() fonksiyonunda okumak istersek, aşağıdaki gibi **iki farklı şekilde** yazabiliriz.
- İki komut arasındaki fark ilk **%d** den sonraki **virgüldür**.

(Kullanıcı tarafından klavyeden girilen sayılar 12 ve 567 olsun)



```
scanf("%d,%d", &X, &Y);
```

/ Sayılar arasına virgül konarak giriş yapılır. Yani klavyede 12,567 yazılır */*

```
scanf("%d %d", &X, &Y);
```

/ Sayılar arasına boşluk konarak giriş yapılır. Yani klavyede 12 567 yazılır */*

C PROGRAMLAMA DİLİ *(Devam)*

main() fonksiyonu

- Şimdiye kadar öğrendiklerimizi uygulamak için...
- **Örnek :** **main()** fonksiyonu içinde **iki tam sayıyı toplayıp ortalamasını alarak** ekrana yazdıran bir C programı yazınız.

C PROGRAMLAMA DİLİ (Devam)

main() fonksiyonu

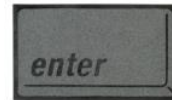
```
main() {  
    int a,b,top ;    /* a ve b toplanacak tam sayılar, top ise toplam degiskeni */  
    float ort ;      /* ortalama degiskeni ort ve tipi kesirli (virgüllü) */  
    printf(" iki sayı giriniz:") ;    /* ekrana iki sayı giriniz: mesajı yaziliyor*/  
    scanf("%d,%d",&a,&b); /* a ve b tam sayı olarak klavyeden okuyacak */  
    top= a+b ;      /* a ve b toplanıp top degiskenine aktariliyor*/  
    ort = top/2 ;   /* ortalama alınıp ort degiskenine aktariliyor */  
    printf (" ortalama= %f dir.",ort) ;    /* ort degeri yaziliyor */  
    }              /* Bu karakterler arasına açıklama yazılabilir */
```


C PROGRAMLAMA DİLİ (Devam)

main() fonksiyonu

- Program çalıştırıldığında **Ekran Girişi ve Çıktısı** ise...
($a=5$ ve $b=4$ olduğu kabul edilsin, **Giriş** sayıların arasına **virgül** konarak kullanıcı tarafından **klavyeden** yapılıyor)

iki sayı giriniz: 5,4
ortalama=4.500000



Not: Çıktıda Float tipi bir sonuç olduğundan noktadan sonra 6 basamak yazıldı.(12 basamak veren derleyiciler de vardır)

C PROGRAMLAMA DİLİ (*Devam*)

main() fonksiyonu

Örnek: Tamsayı olarak verilen **X** ve **Y** nin farkını (**X-Y**) hesaplayıp **SAYILARIN FARKI=** ifadesinden sonra yazdıran **main()** fonk. yazınız.

```
main() { /* Bu main fonksiyonu X ve Y nin farkini hesaplar*/  
    int X,Y,fark ; /* X , Y, fark tam sayi */  
    printf (“ iki sayı giriniz:”) ; /* Ekrana iki sayı giriniz: mesajı yazılsın */  
    scanf (“%d,%d”,&X,&Y) ; /* X ve Y klavyeden okunacak */  
    fark= X-Y ; /* X ve Y çıkarılıp fark değişkenine aktarılıyor*/  
    printf (“ SAYILARIN FARKI= %d”, fark) ; /* çıkış yazılıyor */  
}
```

C PROGRAMLAMA DİLİ *(Devam)*

main() fonksiyonu

- Ekran Girişi ve çıktısı olarak...

(X=25 ve Y=13 olduğu kabul edilirse)

iki sayı giriniz: 25 , 13 

SAYILARIN FARKI= 12

C PROGRAMLAMA DİLİ *(Devam)*

main() fonksiyonu

- **Örnek:** Klavyeden girilen, kesirli sayı olan yarıçap (R) ile dairenin alanı ve çevresini hesaplayarak ekrana yazdıran bir **main()** fonk. yazınız.
($\pi = 3.14$ alınacaktır)

C PROGRAMLAMA DİLİ (Devam)

main() fonksiyonu

```
main () {      /* Bu ornek main fonk. Dairenin alani ve Cevresini hesaplar*/  
    float R,S,C ;      /* R: Yaricap  S: Alan  C: Cevre olsun*/  
    printf( "Yaricapi Giriniz: ");      /* Yaricapi Giriniz: yaz*/  
    scanf ("%f", &R);      /* Yaricapi R degiskenine oku*/  
    S= 3.14 * R * R ;      /* Alani hesapla*/  
    C = 2 * 3.14 * R ;      /* Cevreyi hesapla*/  
    printf ("Dairenin Alani= %f\n", S);      /* Alani ekrana yazdir*/  
    printf ("Dairenin Cevresi= %f", C);      /* Cevreyi yazdir*/  
    }
```

C PROGRAMLAMA DİLİ *(Devam)*

main() fonksiyonu

- Programın **Ekran Girişi ve Çıktısı:**

(Dairenin Yarıçapı = 5.2 kullanıcı tarafından girilmiş olsun)

Yarıçapı Giriniz: 5.2 

Dairenin Alanı= 84.905600

Dairenin Çevresi= 32.656000

C PROGRAMLAMA DİLİ (Devam)

main() fonksiyonu

Örnek: Tabanı (**a**) ve yüksekliği (**h**) olup virgüllü sayı olarak verilen bir üçgenin alanını hesaplayıp yazdıran **main()** fonk. yazınız.

```
main() {                               /* Bu main fonksiyonu üçgenin alanini hesaplar*/  
    float a,h,alan;                     /* a: taban , h: yukseklik ve alan olsun */  
    printf ("Taban ve yuksekligi giriniz:"); /* kenarlari girin  
                                              mesaji */  
    scanf ("%f,%f",&a,&h);             /* a ve h klavyeden okunacak */  
    alan = a*h/2;                       /* alan hesaplanip alan adli degiskene aktariliyor */  
    printf (" Ucgenin Alani= %f ",alan) ; /* cikis yaziliyor */  
}
```

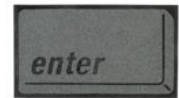
C PROGRAMLAMA DİLİ *(Devam)*

main() fonksiyonu

- Ekran Girişi ve çıktısı:

(Üçgende $a=5.5$ ve $h=4.6$ olarak klavyeden girildi ise...)

Taban ve yüksekliği giriniz: 5.5 , 4.6



Ucgenin Alani= 12.650000 dir.

C PROGRAMLAMA DİLİ *(Devam)*

Formatsız Giriş/Çıkış Komutları

■ Formatsız Giriş ve Çıkış Komutları:

- Çeşitli Formatsız Giriş ve Çıkış komutları bulunmakla birlikte en yaygın kullanılanlar şöyledir:
 - **gets()** : Klavyeden girilen bir karakter dizisini bir değişkene aktarır.
 - **puts()** : Bir karakter dizisini ekrana yazdırır.
 - **getch()** : Klavyeden bir karakter okur ve bunu, ENTER tuşuna basılmaksızın bir değişkene aktarır.
 - **putchar()** : Ekrana bir karakter yazar.
 - **Not:** Bu komutlar için programda **#include<string.h>** başlık dosyası eklenmelidir. Örneklerde gösterilmiyor.

C PROGRAMLAMA DİLİ *(Devam)*

Formatsız Giriş/Çıkış Komutları

- Aşağıdaki örnekte klavyeden girilen bir karakter dizisi ekranda görüntülenmektedir:

```
main{  
    char str[60];  
    printf("Okulunuzun ismi nedir:\n");  
    gets(str);  
    puts(str);  
}
```

C PROGRAMLAMA DİLİ *(Devam)*

Formatsız Giriş/Çıkış Komutları

- Yukarıdaki programın ekran girişi ve çıktısı:

Okulunuzun ismi nedir: 

Teknik Bilimler MYO 

Teknik Bilimler MYO

C PROGRAMLAMA DİLİ *(Devam)*

Sabitler

- **SABİTLER:**
 - Programın yürütülmesi sırasında **değerleri değişmez.**
 - Sabitler genelde **sayısal** ve **alfasayısal** olurlar...
 - Sabitler üzerlerinde **aritmetik işlem yapılarak değiştirilemezler...**
 - **Sayısal sabit...** örn : **3.45**
 - **Alfasayısal sabit...** örn: **"34MYO"**
 - Alfasayısal sabitler **tırnak içerisinde verilir.**
 - Alfasayısal sabit **tek karakter ise tek tırnak, birden fazla karakter ise çift tırnak** içinde verilir.

C PROGRAMLAMA DİLİ (Devam)

Sabitler

- Sabit tanımlanırken **program boyunca değişmeyeceğini** belirtmek için sabit tipinin başına **const** ibaresi konulmalıdır. Meselâ:

```
const int a= - 45;
```

/ a adli ve degeri - 45 olan sabit (tam) sayi demektir , bu tanımdan sonra programda a yazılırsa (- 45) anlamına gelir */*

```
const float b= 2.435298; /* b adli kesirli bir sabit sayi */
```

```
const char ch = 'A'; /* ch adli karsiligi A olan sabit karakter */
```

C PROGRAMLAMA DİLİ (*Devam*)

Değişkenler

■ DEĞİŞKENLER:

- Değişkenler **hafızada saklanan ve program boyunca değeri değişebilen** simgesel isimlerdir.
- C'de temel olarak (*sabitlere benzer*) **Sayısal ve Alfasayısal değişken** türleri bulunmaktadır.
- **Değişkenler içlerine uygun nesnelere konan farklı tipte kutular** olarak düşünülebilir.
- Her nesne her kutuya konamaz. Mesela **ondalık bir sayı, (int) tipi kutuya konmaz**. Konursa veri kayıpları oluşabilir.

C PROGRAMLAMA DİLİ (*Devam*)

Değişkenler

- **Değişkenler tanımlanırken** birden fazla olacak ise **virgül** ile ayrılmalıdırlar.
- Her **ifade/deyim** gibi değişken tanımının sonuna **noktalı virgül konulması** zorunludur.
- **Değişkenlerin hafızada karşılıkları vardır.** Bir **Değişkenin** hafızadaki karşılığı genellikle...
 - Tam sayı
 - Kesirli sayı
 - Karakterolabilir...

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

□ Tam sayı değişkeni tanımlamak için ...

- Örnl:

```
int a,b,c ;    /* 3 tam sayı değişken tanımlanıyor */
```

- Aynı tanımlamayı...

```
int a;
```

```
int b;
```

```
int c;        /* şeklinde de yapabilirdik. */
```

- Burada **a,b,c** (mesela 78 457 -189 -32 0 gibi) değerler alabilir, ama **kesirli sayı olmaz**)

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

□ Kesirli (ondalıklı) sayı değişkeni tanımlamak için...

- Örn:

```
float f=12.345 ;
```

- *Bu **değişkene** ilk değer olarak **12.345** sayısı verilmiştir.*
- *Bu virgüllü sayı (f değişkeni) sabit olmayıp program içinde değişebilecektir.*

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

- **Float tipinde** bir değişken aşağıdaki gibi değerler alabilir:

8.7 0.459 5.45E9 (üslü sayıdır : $5,45 \cdot 10^9$)

- Bu sayıları **int** değişkenine eşitlersek sadece tam kısmını alırız. (8.7 yerine 8 alınır.)
- Ancak **float** değişkenine bir **tam sayı** değişkeni eşitlenebilir. Bu durumda **7** sayısı **7.0** olarak işleme girer.

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

□ Tam sayı ve Float Değişken Tiplerinin Beraber Kullanımı

- Bir **float** değişkenine **int** tipi değişken **eşitlenirse** , C **otomatikman** bu değeri **float tipine çevirir**. Mesela...

```
main() {  
    float a ;  
    int b=10 ;  
    a=b ;  
    printf ("a'nin degeri= %f",a);  
}
```

- Ekran Çıktısı a'nin degeri=10.000000 şeklinde elde edilir.

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

Karakter Tanımlamak için...

- Aşağıda **a** adlı bir **karakter değişkeni** tanımlanmıştır...

```
char a ;
```

- Burada da başlangıç değeri verilebilir...

```
char a='H'      /* Başlangıç değeri H olsun */
```

- **Karakter uzunluğu birden farklıysa** değişken adının yanında bu değer de yazılır.

```
char isim[28]; /* isim adlı uzunluğu 28 olan bir karakter  
değişkeni tanımladık */
```

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

- Bu defa, **a** adlı bir **karakter değişkenine** değer olarak **z** harfini yerleştirerek bu değer karakter tipinde **a'nin degeri=** ifadesinden sonra yazdırıp bir satır atlamayı sağlayan programı **main()** içinde yazalım.

```
main () {  
  
    char a ;  
  
    a= 'z';  
  
    printf ("a'nin degeri= %c\n ",a);  
  
}
```

Ekran Çıktısı: **a'nin degeri= z** şeklinde elde edilir.

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

Örnek: Uyarı mesajından ve klavyeden girilen en fazla 20 karakterli bir ismin okutulmasından sonra Merhaba ifadesinin ardından çıkışta yazdırılma istenmektedir. main() fonk. yazınız.

```
main{  
    char isim[20];  
    printf("İsminizi giriniz: ");  
    scanf("%s",isim);  
    printf ("\n");  
    printf("Merhaba, %s",isim);  
}
```

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

- **Değişken tipleri uzunluklarına göre de değişir. En yaygın olanlar:**
 - **int** : Hafızada **2** bayt (16 bit) yer tutan **tam sayı** için kullanılır
 - **short int** : **int** tipinin **yarısı** uzunluktadır, hafızada **1** bayt yer tutar
 - **long int** : **int** tipinin **iki katı** uzunluktadır, hafızada **4** bayt yer tutar
 - **float** : Tek hassasiyetli kesirli sayı, hafızada **4** bayt yer tutar
*(Normalde virgülden sonra **6** basamak gösterilir)*
 - **double** : Çift hassasiyetli kesirli sayı, hafızada **8** bayt yer tutar
*(Normalde virgülden sonra **12** basamak gösterilir)*
 - **char** : Tek yada daha fazla karakter saklar.
*(Her karakter hafızada **1** bayt yer tutar)*

C PROGRAMLAMA DİLİ *(Devam)*

Değişkenler

- Bazı değişken tanımlama örnekleri şöyle olabilir:

```
int x=69,y,z;   /* x 'in başlangıç değeri 69, y ve z tam sayı */
```

```
short int say;   /* say adlı değişken */
```

```
long int sonuc; /* sonuc adlı uzun tam sayı değişkeni */
```

```
float pi;       /* pi adlı kesirli bir değişken */
```

```
double e= 2.711828182845;
```

```
/* e adlı çift hassasiyetli kesirli sayı değişkeni başlangıç değeri ile  
tanımlanmıştır */
```

```
char sehir[15];   /* sehir adlı 15 uzunluklu karakter */
```


C PROGRAMLAMA DİLİ *(Devam)*

İşaretsiz Değişkenler

□ İşaretsiz Değişkenler:

- Bir başka tip tanımlaması da **sayının işaretsiz olduğunu** belirtir. Bunun için **unsigned** kelimesi kullanılır. Mesela;

unsigned int sayi ;

Burada **sayi** adlı değişkenin **tam sayı türünde** olduğu, **sadece pozitif sayı olacağı** anlaşılır.

- Böylece çalışma alanı sayı **0 ile 65535 aralığında** olur. *(int 16 bit idi. $2^{16} = 65536$ farklı sayı; sıfır dahil!)*

C PROGRAMLAMA DİLİ (Devam)

İşaretsiz Değişkenler

unsigned int sayi ; /* sayi adlı sadece **pozitif** 16 bit*/

unsigned long sayi ; /* sayi adlı sadece **pozitif** 32 bit */

- **unsigned** (işaretsiz) ifadesi olmasaydı **çalışma aralığı...**

-327680..... +32767 olacaktı.

- Yani **sayi işaretili olacak ve hem pozitif hem negatif değerler** alabilecek, **65535** sayısı yarı yarıya paylaşılacaktı.

NOT: *Mikrodenetleyiciler dersindeki pozitif (+) ve negatif (-) değerler alabilen **işaretili sayıları** hatırlayınız.*

C PROGRAMLAMA DİLİ (Devam)

Değişkenler

Sık kullanılan Değişken Türlerini bir Tablo halinde Özet olarak tekrar verirsek...

Tür Adı	Tanım ve Açıklama
int	Tamsayı değişken tanımlama için kullanılır.
Unsigned int	İşaretsiz yani Pozitif Tamsayı değişken tanımlama için kullanılır.
long int	Uzun Tamsayı (Yani daha büyük tam sayılar) değişken tanımlama için kullanılır.
Unsigned long int	İşaretsiz uzun Tamsayı değişken tanımlama için kullanılır.
Float	Kesirli değişken tanımlama için kullanılır.
Double	Kesirli çift duyarlıklı değişken tanımlama için kullanılır.

C PROGRAMLAMA DİLİ (Devam)

Fonksiyonlar

❖ C FONKSİYONLARI

- İyi tasarlanmış bir program, büyük bir **programı parçalara (fonksiyonlara)** böler.
- **Fonksiyon belli bir işi yapan program parçasıdır.**
- **main()** de bir fonksiyondur. Ancak daha önce belirtildiği gibi olmazsa olmazdır ve program **her zaman buradan çalışmaya başlar.**

C PROGRAMLAMA DİLİ (Devam)

Fonksiyonlar

- Fonksiyonlar **değer (parametre) kullanarak ya da kullanmadan** çalışırlar.
- Bir fonksiyon yazılıp hatalardan arındırıldıktan sonra **başka programlarda da** kullanılabilir.
- **Kütüphane fonksiyonları** da birer hazır fonksiyondur, daha önce yazılıp yerleştirilmişlerdir ve derleme sırasında kullanılırlar.

C PROGRAMLAMA DİLİ (Devam)

Fonksiyonlar

- **Bir değer alıp işlem yapan ama kendisi bir değer vermeyen (üretmeyen) bir fonksiyon olarak...**

```
printf("Merhaba, Dünya");
```

örnek olarak verilebilir.

- Burada printf() fonksiyonuna, **Merhaba, Dünya** değer olarak girmekte ancak kendisi **bir değer üretip başka fonksiyona göndermemektedir.**
- **printf()** sadece **ekrana çıktı yazmaktadır.**

C PROGRAMLAMA DİLİ (Devam)

Fonksiyonlar

- Standart C 'de çok sayıda (farklı başlık dosyalarıyla kullanılan) kütüphane(hazır) **fonksiyonları** bulunmakla birlikte şimdilik tek bir örneği inceleyelim.
- Aşağıdaki örnek kütüphane fonksiyonu, büyük harfle yazılmış bir karakteri küçük harfe çevirmektedir.

C PROGRAMLAMA DİLİ (Devam)

tolower() Fonksiyonu

□ **tolower()** Fonksiyonu

- Bu fonksiyona parantez içinde parametre olarak **büyük harf girer ve küçük harf şeklinde bir değer üretir.**
- **Örnek: main()** fonk. içinde **tolower()** fonksiyonu kullanarak büyük **K** karakterini küçük **k** 'ya dönüştürüp yazdıran bir program yazınız.

C PROGRAMLAMA DİLİ *(Devam)*

tolower() Fonksiyonu

```
main() {  
    char a,b;  
    b= 'K' ;  
    a= tolower(b) ;  
    printf (" K'nin kucugu= %c",a);  
}
```

Ekran Çıktısı:

K'nin kucugu=k

elde edilecektir.

C PROGRAMLAMA DİLİ *(Devam)*

Fonksiyonlar

- Önceki **tolower()** fonksiyonuna **K** değerini **doğrudan** değer olarak vermemiz de mümkündür. Aşağıdaki örnekte **b** değişkeni yoktur, doğrudan **'K'** değeri fonksiyona parametre olarak verilmektedir. Çıktı önceki örnekle **aynı** olacaktır.

```
main() {  
    char a ;  
    a= tolower('K') ;  
    printf ("K'nin kucugu = %c",a) ;  
}
```

Not: *Tersine* küçük karakteri büyük karaktere dönüştürmek için benzer şekilde **toupper()** fonksiyonu kullanılabilir.