

Doğu Akdeniz Üniversitesi  
Bilgisayar ve Teknoloji Yüksek Okulu  
Bilgisayar Programcılığı Bölümü

**BTEP 102 – Veri Yapıları ve Programlama**

**Bölüm 1: C Programlama Diline Giriş**



R. KANSOY

1

**Konu Başlıkları**


**1. C Programlama Diline Giriş**

- 1.1 C Programlarının Yapısı
- 1.2 Main() Fonksiyonu
- 1.3 C Ön İşlemcisi
- 1.4 Değişkenler ve Temel Veri Türleri
- 1.5 Sabitler (Değişmezler)
- 1.6 Basit Veri Giriş Çıkışları
- 1.7 Operatörler (İşleçler)

BTEP 102 – Veri Yapıları ve Programlama

2/33

2




## 1.1 C Programlarının Yapısı

- ✓ Bir C programı bir veya daha fazla **fonksiyondan** oluşur.
- ✓ Her bir fonksiyon bir veya daha fazla sayıda **deyim/ifade** içerir.
- ✓ Programın amacına ulaşması için gereken işlemleri yerine getiren komutlara **deyimler/ifadeler** denilir.
- ✓ Çalıştırılacak ifadeler, **main ()** fonksiyonu içerisine **{ }** işaretleri arasına yazılır ve **(;)** işareti ile son bulur.
- ✓ Bu iki işaret **{ }** arasında kalan kısımlara **blok** ismi verilir.
- ✓ Bir program içerisinde kullanılacak değişkenlerin adlarını belirlemek amacı ile yazılan **bildirimler** de bu kısımda bulunur .

BTEP 102 – Veri Yapıları ve Programlama

3/33

3



## C Programlarının Yapısı - devam...

- ✓ C'de bir fonksiyon altıyordam (subroutine) olarak da adlandırılır ve bunlar programın bir başka yerinden isimleri kullanılarak çağrılabilir.

**Kod 1.1 Basit Bir C Programının Görünümü**

```

/* ilk C programı */
#include <stdio.h>

main()
{
    printf("Merhaba BTEP102 Öğrencileri");
}

```

Önişlemci Direktifi ← # include <stdio.h> → Başlık Dosyası

main() → Ana Fonksiyon


{ } → Blok

- ✓ C programının herhangi bir yerine açıklama satırı eklemek için açıklamanın başına **(/\*)** işaretleri sonuna ise **(\*/)** işaretleri eklenir.
- ✓ Bu işaretler arasına kaydedilen ifadeler C derleyicisi tarafından “yok” sayılır. Yani herhangi bir işleme tabi tutulmaz.

BTEP 102 – Veri Yapıları ve Programlama

4/33

4




## 1.2 Main() Fonksiyonu

- ✓ Bir C programı bir veya daha fazla fonksiyondan oluşur.
- ✓ Bir C programının içinde çok sayıda fonksiyon yer alabilir ancak mutlaka bir `main()` fonksiyonu bulunmalıdır.
- ✓ İlk çalıştırılacak olan fonksiyon `main()` yani ana fonksiyondur.
- ✓ C'de programcı tarafından tanımlanan fonksiyonlar dışında, hazır kitaplık fonksiyonları da bulunmaktadır. Ör: `printf()`

BTEP 102 – Veri Yapıları ve Programlama

5/33

5




## 1.3 C Önışlemcisi

- ✓ C programları kendi derleyicisi ile ilişkin halindedir. Bu ilişki C önışlemcisi yardımıyla sağlanır.
- ✓ Önışlemciler çeşitli emirlerden oluşabilir.
- ✓ Bu emirler C derleyicisinin kaynak kodunu denetlemekte kullanılır.
- ✓ Önışlemci emirleri C programı içerisinde (#) işareti ile başlar ve (;) işareti ile sonlandırılmaz.
- ✓ C'nin en çok kullanılan önışlemci emirleri `#include` ve `#define` ile tanımlanmaktadır.

BTEP 102 – Veri Yapıları ve Programlama

6/33

6




## C Önışlemcisi - devam...

- ✓ C programlama dilinde belirli işleri gerçekleştiren program parçaları C' nin temel yapısı içinde yer almaz.
- ✓ Bu nedenle program içinde kullanılan fonksiyonların kendilerinin ya da başlıklarının yer aldığı dosyalar **#include** emri kullanılarak ana programa dahil edilir.
- ✓ C programına dahil edilen **.h** uzantılı bu dosyalara **başlık (Header)** dosyası adı verilir.
- ✓ Standart kitaplık fonksiyonları hakkındaki bilgiler, bu tür başlık dosyaları içinde yer alır.
- ✓ Örneğin **printf** fonksiyonunun özet bilgisi **stdio.h** isimli dosyanın içinde yeri almaktadır.
- ✓ Bir programda **printf** fonksiyonu kullanılacaksa, programın başında **stdio.h** (Standart Input Output) dosyasının **#include** emri ile programa dahil edilmesi gerekir.
- ✓ Bu dosya programa **#include <stdio.h>** direktifi ile dahil edilir.

BTEP 102 – Veri Yapıları ve Programlama

7/33

7



## C Önışlemcisi - devam...

- ✓ Program içinde kullanılacak değişmezlerin tanımlanması için **#define** önışlemci emri kullanılır.

**Kod 1.2 : Sembolik Değişmezlerin (sabitlerin) Tanımlanması**

```
# include <stdio.h>


# define SON 50

main()
{
    deyimler;
}
```

BTEP 102 – Veri Yapıları ve Programlama

8/33

8



## C Önışlemcisi - devam...

### C'deki bazı kütüphaneler;


**TABLO 1.1 C'deki Bazı Kütüphaneler**

| KÜTÜPHANE  | İÇERİĞİ                                      |
|------------|--|
| stdio.h    | Standart giriş-çıkış komutları               |
| conio.h    | DOS destekli giriş-çıkış komutları           |
| math.h     | Matematiksel fonksiyonlar                    |
| stdlib.h   | Dönüşüm, sıralama, arama vb. komutları       |
| graphics.h | Grafik ortam komutları                       |
| dos.h      | DOS fonksiyonları                            |
| ctype.h    | Karakter dönüşüm ve sınıflandırma komutları  |
| string.h   | Alfasayısal ve bazı bellek yönetim komutları |

BTEP 102 – Veri Yapıları ve Programlama

9/33

9




## 1.4 Temel Veri Türleri ve Değişkenler

- ✓ Oluşturulan programın kendi iç işlemlerini yaparken, veri girişlerini aldığı ve üzerinden kullanıcılara sonuçları gönderdiği, program elemanlarına değişken adı verilir.
- ✓ Programın, değişkenleri program esnasında kullanabilmesi için değişkenlerin programın başında veya fonksiyon başlarında tanıtılması gerekmektedir.
- ✓ Değişken tanımlama işlemi ile:
  - Derleyiciye program içerisinde kullanılacak tüm değişkenlerin isimlerini bildirilir.
  - Program içinde kullanım amacına göre değişkenin tipi belirlenir. Değişken bir **veri tipi** almak zorundadır.
  - Değişkenin hafıza üzerinde kullanacağı alan da seçilen **veri tipi** ile belirlenecektir.

BTEP 102 – Veri Yapıları ve Programlama

10/33

10



## Temel Veri Türleri ve Değişkenler - dvm...


**TABLO 1.2 C Programlama Diline İlişkin Temel Veri Türleri**

| Veri Türü | Anlamı                                |
|-----------|---------------------------------------|
| char      | Karakter veriler                      |
| int       | Tamsayı veriler                       |
| float     | Tek duyarlıklı kayan noktalı sayılar  |
| double    | Çift duyarlıklı kayan noktalı sayılar |
| void      | Değer içermeyen                       |

BTEP 102 – Veri Yapıları ve Programlama

11/33

11



## Temel Veri Türleri ve Değişkenler - dvm...

### Tamsayılar → int


- ✓ Pozitif ve negatiflik kriterine göre
  - int → işaretli tamsayı (hem pozitif hem negatif olabilir)
  - unsigned int → işaretsiz tamsayı (yalnızca pozitif)
- ✓ Hafızada kapladığı yere göre
  - short → kısa tamsayı (Hafızada 2 byte ile temsil edilir)
  - long → uzun tamsayı (Hafızada 4 byte ile temsil edilir)

| Tür ismi           | Uzunluk | Sınır Değerleri              |
|--------------------|---------|------------------------------|
| signed int         | 2       | -32.768 32.767               |
|                    | (4)     | -2.147.483.648 2.147.483.647 |
| unsigned int       | 2       | 0 65.535                     |
|                    | (4)     | 0 4.294.967.295              |
| short int          | 2       | -32.768 32.767               |
| short unsigned int | 2       | 0 65.535                     |
| long int           | 4       | -2.147.483.648 2.147.483.647 |
| long unsigned int  | 4       | 0 4.294.967.295              |

BTEP 102 – Veri Yapıları ve Programlama

12/33

12



## Temel Veri Türleri ve Değişkenler - dvm...

### Gerçek Sayılar


- ✓ Hafızada kapladığı yere göre
  - float → 4 byte uzunluğunda
  - double → 8 byte uzunluğunda
  - long double → 10 byte uzunluğunda
- ✓ long, short, signed, unsigned float ile birlikte kullanılmaz.
- ✓ float tipi noktadan sonra 6 rakam depolarken, double tipi (long veya short) 12 rakam depolar.

| Tür ismi    | Uzunluk | Sınır Değerleri   |
|-------------|---------|---|
| float       | 4       | $-10^{-38} \dots -10^{38}$ $10^{-38} \dots 10^{38}$         |
| double      | 8       | $-10^{-308} \dots -10^{308}$ $10^{-308} \dots 10^{308}$     |
| long double | 10      | $-10^{-4932} \dots -10^{4932}$ $10^{-4932} \dots 10^{4932}$ |

BTEP 102 – Veri Yapıları ve Programlama

13/33

13



## Temel Veri Türleri ve Değişkenler - dvm...


### Harfler

- ✓ Depolayabildiği değere yere göre
  - char (veya signed char) 1 byte
    - -128...127 arasında değer alır.
  - Unsigned char 1 byte
    - 0...255 arasında değer alır.
- ✓ Tek karakter depolamak için kullanılır.
- ✓ Küçük harf büyük harf ayrı değerlendirilir.

BTEP 102 – Veri Yapıları ve Programlama

14/33

14



## Temel Veri Türleri ve Değişkenler - dvm...

### VOID


Fonksiyonlar tarafından kullanılır, değer olmaması anlamına gelir:

- void fonksiyon1 (void)
  - Ne değer alır ne de değer döndürür.
- int fonksiyon1 (void)
  - Bir deger almaz ama fonksiyonun basari ile tamamlandigini gösteren bir deger döndürür.
- void fonksiyon1(int)
  - Çalışırken üst fonksiyondan değer alır, ama fonksiyonun tamamlandığını gösteren bir değer döndürmez.

BTEP 102 – Veri Yapıları ve Programlama

15/33

15



## Temel Veri Türleri ve Değişkenler - dvm...

### Değişkenleri İsimlendirme Kuralları


- ✓ C'de bütün degiskenler bir harf veya altçizgi karakteri (\_) ile baslamak zorundadır.
- ✓ İlk karakterden sonra harfler, rakamlar veya altçizgi karakteri (\_) gelebilir. Özel karakterler (#,\$,@,%,&,...) ve Türkçe karakterler (ş,g,ı,ş,ç,ü,...) hiçbir sekilde degisken isimlendirme de kullanilamaz.
- ✓ Degisken adının ilk 31 karakteri degerlendirilir. 31. karakterden sonrakiler dikkate alınmaz.
- ✓ Büyük harf, küçük harfle aynı degildir.
  - int deger
  - int Deger ayrı ayrı degiskenler olarak algılanır.
- ✓ C'nin anahtar kelimeleri degisken adı olarak kullanilamaz.
- ✓ Sabitler'in isimlendirmesinde de degiskenlerin isimlendirilmesi kuralları geçerlidir. Fakat, sabitlerin isimlendirilmesinde yaygın olarak büyük harf kullanılmaktadır.

BTEP 102 – Veri Yapıları ve Programlama

16/33

16





## Temel Veri Türleri ve Değişkenler - dvm...


### Değişken Veri Türünün Bildirimi

- ✓ C programı içinde kullanılacak bir değişkenin veri türünü bildirmek için şu şekilde bir tanım yapılır: veri türü değişken adı;
  - int mil;
  - float km;
  - char ilk\_harf;
- ✓ Değişken bildirimi yapıldığı esnada, değişkene ilk değer ataması yapılabilir.
  - float PI=3.14;
  - float ara\_toplam=100;

BTEP 102 – Veri Yapıları ve Programlama

17/33

17



## Temel Veri Türleri ve Değişkenler - dvm...

### Değişken Türleri

- ✓ C programı içinde farklı amaçlara yönelik değişken tanımları yapılabilir. Değişken türlerini şu şekilde sıralayabiliriz:
  - Yerel değişkenler
  - Küresel değişkenler
  - extern değişkenler
  - static değişkenler
  - auto değişkenler
  - register değişkenler

BTEP 102 – Veri Yapıları ve Programlama

18/33

18

## Temel Veri Türleri ve Değişkenler - dvm...

### YEREL DEĞİŞKENLER

- ✓ Değişken veri türü bildirimleri bir fonksiyonun içinde yada dışında yapılabilir.
- ✓ Program içinde birden fazla fonksiyon varsa, sadece tanımlandığı fonksiyonda geçerli olabilecek değişkenlere **yerel değişken** (local variable) adı verilir.

**Kod 1.3 : Yerel Değişkenlerin Kullanımı**

```
# include <stdio.h>

main()
{
    int i=5;
    printf("sayı değeri 1: %d\n",i );
    fonk1();
}

fonk1();
{
    int i;
    printf("sayı değeri 2: %d",i );
}
```

**main() içindeki i değişkeninin değeri fonk1() içinde geçerli değildir.**

BTEP 102 – Veri Yapıları ve Programlama

19/33

19

## Temel Veri Türleri ve Değişkenler - dvm...

### KÜRESEL DEĞİŞKENLER

- ✓ Eğer bir değişkenin program içindeki tüm fonksiyonlar için geçerli olması isteniyorsa, değişken bildirimi tüm fonksiyonların dışında yapılır.
- ✓ Bu tür değişkenlere **küresel değişken** (global variable) adı verilir.

**Kod 1.4 : Yerel Değişkenlerin Kullanımı**

```
# include <stdio.h>
int i=5;
main()
{
    printf("sayı değeri 1: %d\n",i );
    fonk1();
}


fonk1();
{
    int i;
    printf("sayı değeri 2: %d",i );
}
```

**Fonksiyonların dışındaki i değişkeninin değeri tüm fonksiyonlar için geçerlidir.**

BTEP 102 – Veri Yapıları ve Programlama

20/33

20




## 1.5 Sabitler (Değişmezler)

- ✓ Değişmez yada bir başka deyişle sabit, değeri değişmeyen program bileşenleridir.
- ✓ C programlarında aşağıda belirtilen veri türlerine sahip değişmezler yer alabilir:
  - Tamsayı değişmezler
  - Kayan noktalı değişmezler
  - Karakter değişmezler
  - Karakter dizisi değişmezler
- ✓ Bir değişmez C programı içinde const sözcüğü ile tanımlanır.
  - `const sabit adı = değeri ;`
  - `char const erkek='E';`
  - `int const standart=120;`
  - `float const x=2.2321;`
- ✓ Sabit değer ataması yapılırken değer ataması zorunludur. Sabit değerlerin içeriği programın çalışması esnasında değiştirilemez.

21/33

21




## 1.6 Basit Veri Giriş Çıkışları

- ✓ Bir C programı içinde değişkenlere değerler atayarak, bu değerler üzerinde çeşitli işlemler yapılabilir ancak bu tür veriler , program hazırlanırken program içine gömülen değerlerdir.
- ✓ Programın yürütülmesi esnasında da programa veri aktarmak gerekebilecektir.
- ✓ C'de yaygın olarak kullanılan veri giriş ve çıkış fonksiyonları, C standart giriş-çıkış kütüphanesinde bulunur.
  - `#include <stdio.h>`
- ✓ Bilgisayarın standart giriş birimi klavye, çıkış birimi ise ekran olarak kabul edilmektedir.

22/33

22



## Basit Veri Giriş Çıkışları - dvm...

### printf Fonksiyonu

- ✓ Değişkenlerin değerlerini, hesaplanan sonuçları ya da mesajları ekranda göstermek için kullanılır.
- ✓ Bu fonksiyon şu şekilde tanımlanmaktadır:
  - `printf(kontrol karakterleri, değişken listesi);`
- ✓ Kontrol karakterleri, çıkışı yapılacak argümanların biçimlendirilmesini sağlar. Bu tür biçimlendirme ifadeleri % işareti ile başlar.

Parametreler (arguments)

```
printf( "Hesaplanan yaşınız %d dir\n", yas );
```

↓  
Fonksiyon adı

↓  
Çıktı metin (string) formatı

↓  
Değişken listesi(print list)

Hesaplanan yaşınız 17 dir

BTEP 102 – Veri Yapıları ve Programlama
23/33

23



## Basit Veri Giriş Çıkışları - dvm...

**TABLO 1.3 % İşareti ile Birlikte Kullanılan Kontroller**

| Veri Türü | Anlamı                      |
|-----------|-----------------------------|
| c         | İşaretsiz bir karakter      |
| s         | Karakter dizisi             |
| d,i       | İşaretili bir ondalık sayı  |
| u         | İşaretsiz bir ondalık sayı  |
| o         | İşaretsiz bir sekizlik sayı |
| x         | İşaretsiz bir onaltılı sayı |
| e         | Çift duyarlılık sayı        |

**TABLO 1.4 Escape (\) İşareti ile Birlikte Kullanılan Kontroller**

| Kontrol | Anlamı                             |
|---------|------------------------------------|
| \n      | Satır atlama                       |
| \t      | Yatay sekem                        |
| \a      | İkaz (alert). Beep sesi            |
| \\      | Ters bölü işaretini yazdırmak için |
| \"      | Tırnak işaretini yazdırmak için    |
| \b      | İmleci bir sola kaydırma           |
| \f      | Sayfa atlama                       |
| \r      | Satır başı                         |

BTEP 102 – Veri Yapıları ve Programlama
24/33

24



## Basit Veri Giriş Çıkışları - dvm...

### KOD1.5 Farklı Türlerdeki Veri Çıkışları

```
# include <stdio.h>

main()
{
    char k[]="test";
    float s1;
    double s2;

    s1=123;
    s2=607.291;

    printf("Karakter: %s\n",k);
    printf("1. sayi: %f\n",s1);
    printf("2. sayi: %e\n",s2);
}

Sonuç:
Karakter: test
1.sayi : 123.000000
2.Sayi: 6.072910e+002
```

25/33

25



## Basit Veri Giriş Çıkışları - dvm...

### scanf Fonksiyonu

- ✓ Klavyeden girilen verileri programa aktarmak için C standart kitaplığının scanf() fonksiyonu kullanılır.
- ✓ Bu fonksiyon şu şekilde tanımlanmaktadır:
  - **scanf(kontrol karakterleri, değişken listesi);**
- ✓ Dışarıdan değer girilecek bütün değişkenlerin başına & (ampersand) işareti konulur.

Parametreler (arguments)

**scanf( "%d%f", &yas,&kilo );**

↓

Fonksiyon adı

↓

Girdi metin (string) formatı

↓

Değişken listesi(input list)

26/33

26



## Basit Veri Giriş Çıkışları - dvm...

**KOD1. 6 Klavyeden Veri Girişi**

```
# include <stdio.h>

main()
{
    int sayi;

    printf("Bir Sayı Giriniz:");
    scanf("%d",&sayi);
    printf("%d",sayi);
}
```

**Mesaj**


**Klavyeden bir değer giriliyor.**

**Klavyeden girilen değer görüntüleniyor**

BTEP 102 – Veri Yapıları ve Programlama

27/33

27




## 1.7 Operatörler (İşleçler)

- ✓ Aritmetik işlemler başta olmak üzere, büyüklüklerin karşılaştırılması ve mantıksal karşılaştırmaların yapılmasını sağlamak için operatörler kullanılır.
- ✓ Operatörler değişkenler ve sabitler üzerinde işlem yapabilmemize olanak verir.
- ✓ C programlarında en sık kullanılan operatörler;
  - Aritmetik Operatörler
  - Karşılaştırma Operatörleri
  - Mantıksal Operatörler

BTEP 102 – Veri Yapıları ve Programlama

28/33

28



Aritmetik Operatörler


✓ Toplama, çıkarma, çarpma ve bölme işlemlerini yapmaya yönelik kullanılan operatörlerdir.

| Operatör | Anlamı                | Aritmetik ifade | C'deki karşılığı |
|----------|-----------------------|-----------------|------------------|
| +        | Toplama               | $b + c_1$       | $b + c1$         |
| -        | Çıkarma               | $ax + y$        | $a * x + y$      |
| *        | Çarpma                | $2a$            | $2 * a$          |
| /        | Bölme                 | $z(x+y)/2$      | $z * (x+y) / 2$  |
| %        | Bölme işleminde kalan |                 |                  |
| ++       | Bir artırma           |                 |                  |
| --       | Bir azaltma           |                 |                  |

BTEP 102 – Veri Yapıları ve Programlama

29/33

29



Aritmetik Operatörler - devam...

| Kod 1.7 : ++ ve -- Operatörlerinin Kullanımı   | Kod 1.8 : ++ ve -- Operatörlerinin Kullanımı  | Kod 1.9 : ++ ve -- Operatörlerinin Kullanımı  |
|--|---|---|
| <pre># include &lt;stdio.h&gt;  main() {     int a=50;     int b=30;      printf("%d\n",a++ );     printf("%d",++b ); }  Sonuç: 50        31</pre> | <pre># include &lt;stdio.h&gt;  main() {     int a=50;     int b=30;      a++;     ++b;      printf("%d\n",a );     printf("%d",b ); }  Sonuç: 51        31</pre> | <pre># include &lt;stdio.h&gt;  main() {     int a=50;     int b=30;     int x,y;      x=a++;     y=++b;      printf("%d\n",x);     printf("%d",y ); }  Sonuç: 50        31</pre> |

BTEP 102 – Veri Yapıları ve Programlama

30/33

30



## Karşılaştırma Operatörleri

✓ İki sayısal değeri yada iki karakteri karşılaştırmak amacıyla kullanılır.

| OPERATÖR | ANLAMI          |
|----------|-----------------|
| ==       | Eşit            |
| !=       | Eşit değil      |
| >        | Büyük           |
| <        | Küçük           |
| >=       | Büyük veya Eşit |
| <=       | Küçük veya Eşit |


7 == 3 //yanlış sonucunu verir.  
 7 == 7 //dogru sonucunu verir.  
 7 != 3 //dogru sonucunu verir.  
 7 > 3 //dogru sonucunu verir.  
 7 < 3 //yanlış sonucunu verir.  
 7 >= 7 //dogru sonucunu verir.  
 7 <= 7 //dogru sonucunu verir.  
 7 <= 9 //dogru sonucunu verir.

x\*y > z  
 x == 10  
 x + 3 <= 10

BTEP 102 – Veri Yapıları ve Programlama

31/33

31



## Mantıksal Operatörler

✓ İki veya daha fazla sayıdaki koşulun birlikte sınanması amacıyla kullanılır.

| OPERATÖR | ANLAMI |
|----------|--------|
| &&       | Ve     |
|          | Veya   |
| !        | Değil  |

!true //yanlış sonucunu üretir.  
 !false //dogru sonucunu üretir.  
 !(7<3) //dogru sonucunu üretir.


| x | y | x değil | x ve y | x veya y |
|---|---|---------|--------|----------|
| 1 | 1 | 0       | 1      | 1        |
| 1 | 0 | 0       | 0      | 1        |
| 0 | 1 | 1       | 0      | 1        |
| 0 | 0 | 1       | 0      | 0        |

BTEP 102 – Veri Yapıları ve Programlama

32/33

32





## Mantıksal Operatörler - dvm.

**KOD 1.10 Mantıksal Operatörlerin Kullanımı**

```
# include <stdio.h>

main()
{   int sayi1,sayi2;

    printf("Birinci sayıyı giriniz:");
    scanf("%d", &sayi1);

    printf("İkinci sayıyı giriniz:");
    scanf("%d", &sayi2);

    if (sayi1>0 && sayi2>0)
        printf("Her iki sayı pozitif");
    else
        printf("Sayılardan en az biri negatif");
}
```

Bu program yürütülmeye başladıktan sonra, klavye yardımıyla -5 ve 15 sayısal değerleri girildiğini varsayarsak, "Sayılardan en az biri negatif" mesajı görüntülenir.

BTEP 102 – Veri Yapıları ve Programlama

33/33