

Konu Başlıkları 1. C Programlama Diline Giriş 1.1 C Programlarının Yapısı 1.2 Main() Fonksiyonu 1.3 C Önişlemcisi 1.4 Değişkenler ve Temel Veri Türleri 1.5 Sabitler (Değişmezler) 1.6 Basit Veri Giriş Çıkışları 1.7 Operatörler (İşleçler)



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 yazılıan bildiriler de bu kısımda bulunur .

3/33

3

C Programlarının Yapısı - devam...

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

Kod 1.1 Basit Bir C Programının Görünümü

/* ilk C programı */ → Açıklama Satırı

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

main() → Ana Fonksiyon
{
 printf("Merhaba BTEP102 Öğrencileri"); }

- 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.

4/33

Blok



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()

5/33

5



1.3 C Önişlemcisi

- ✓ C programları kendi derleyicisi ile ilişkin halindedir. Bu ilişki C önişlemcisi yardımıyla sağlanır.
- ✓ Önişlemciler çeşitli emirlerden oluşabilir.
- ✓ Bu emirler C derleyicisinin kaynak kodunu denetlemekte kullanılır.
- ✓ Önişlemci emirleri C programı içerisinde (#) işareti ile başlar ve (;) işareti ile <u>sonlandırılmaz</u>.
- ✓ C'nin en çok kullanılan önişlemci emirleri #include ve #define ile tanımlanmaktadır.

6/33

6

BTEP 102 – Veri Yapıları ve Program



C Önişlemcisi - 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.

7/33

/



C Önişlemcisi - devam...

✓ Program içinde kullanılacak değişmezlerin tanımlanması için #define önişlemci emri kullanılır.

Kod 1.2 : Sembolik Değişmezlerin (sabitlerin) Tanımlanması
include <stdio.h>
define SON 50
main()
{
 deyimler;
}

8/33

R

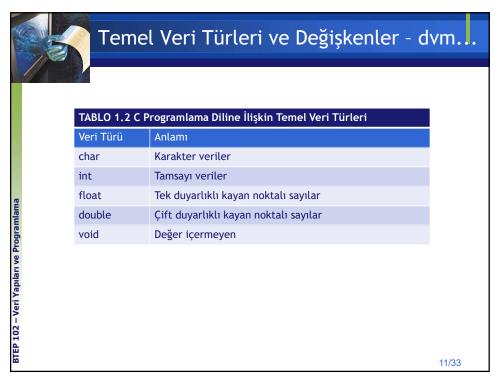




1.4 Temel Veri Türleri ve Değişkenler

- ✓ Oluşturulan programın kendi iç islemlerini yaparken,veri girişlerini aldığı ve üzerinden kullanıcılara sonuçları gönderdiği, program elemanlarına degişken adı verilir.
- ✓ Programın, degiskenleri program esnasında kullanabilmesi için degiskenlerin programın basında veya fonksiyon baslarında tanıtılması gerekmektedir.
- ✓ Degisken tanımlama işlemi ile:
 - Derleyiciye program içerisinde kullanılacak tüm degiskenlerin isimlerini bildirilir.
 - Program içinde kullanım amacına göre degiskenin tipi belirlenir. Degisken bir veri tipi almak zorundadır.
 - Degiskenin hafıza üzerinde kullanacagı alan da seçilen veri tipi ile belirlenecektir.

10/33



Temel Veri Türleri ve Değişkenler - dvm... Tamsayılar → int Pozitif ve negatiflik kriterine göre int → isaretli tamsayı (hem pozitif hem negatif olabilir) unsigned int → isaretsiz tamsayı (yalnızca pozitif) Hafızada kapladıgı yere göre short → kısa tamsayı (Hafızada 2 byte ile temsil edilir) long → uzun tamsayı (Hafızada 4 byte ile temsil edilir) BTEP 102 – Veri Yapıları ve Programl Tür ismi Uzunluk Sınır Değerleri 32.767 -32.768 signed int (4) -2.147.483.648 2.147.483.647 2 65.535 unsigned int (4) 4.294.967.295 short int 32.767 -32,768 2 short unsigned int 2 65,535 2.147.483.647 -2.147.483.648 long int long unsigned int 4 294 967 295

12/33



Gerçel Sayılar

- √ Hafızada kapladıgı 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 10 ³⁸	10-38 1038
double	8	-10 ⁻³⁰⁸ 10 ³⁰⁸	10-308 10308
long double	10	-10-4932104932	10-4932104932

13/33

13

Temel Veri Türleri ve Değişkenler - dvm...

Harfler

- ✓ Depolayabildigi değere yere göre
 - char (veya signed char) 1 byte
 - -128...127 arasında deger alır.
 - Unsigned char 1 byte
 - 0...255 arasında deger alır.
- ✓ Tek karakter depolamak için kullanılır.
- \checkmark Küçük harf büyük harf ayrı degerlendirilir.

14/33



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 basarı ile tamamlandıgını gösteren bir deger döndürür.
- void fonksiyon1(int)
 - Çalısırken üst fonksiyondan değer alır, ama fonksiyonun tamamlandığını gösteren bir değer döndürmez.

15/33

15

Te

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 (s,g,1,S,ç,ü,...) hiçbir sekilde degisken isimlendirme de kullanılamaz.
- ✓ 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 kullanılamaz.
- ✓ Sabitler'in isimlendirmesinde de degiskenlerin isimlendirilmesi kuralları geçerlidir. Fakat, sabitlerin isimlendirilmesinde yaygın olarak büyük harf kullanılmaktadır.

16/33

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 değişen adı;
 - int mil;
 - float km;
 - char ilk_harf;
- ✓ Degisken bildirimi yapıldıgı esnada, degiskene ilk deger ataması yapılabilir.
 - float PI=3.14;
 - float ara toplam=100;

17/33

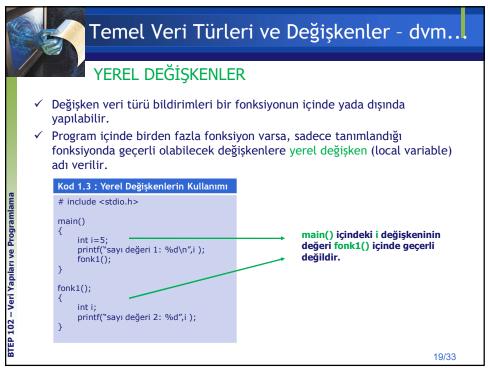
17

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 değikenler
 - register değişkenler

18/33



Temel Veri Türleri ve Değişkenler - dvm... KÜRESEL DEĞİŞKENLER ✓ Eğer bir değişkenin program içindeki tüm fonksiyonlar için geçerli olması isteniyorsa, değişken bildirimi tüm foksiyonların dışında yapılır. ✓ Bu tür değişkenlere küresel değişken (global variable) adı verilir. Kod 1.4 : Yerel Değişkenlerin Kullanımı # include <stdio.h> int i=5; main() Fonksiyonların dışındaki i printf("sayı değeri 1: %d\n",i); değişkeninin değeri tüm fonk1(); fonksiyonlar için geçerlidir. } fonk1(); printf("sayı değeri 2: %d",i); 20/33



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 deger ataması yapılırken deger ataması zorunludur. Sabit degerin içerigi programın çalısması esnasında degistirilemez.

21/33

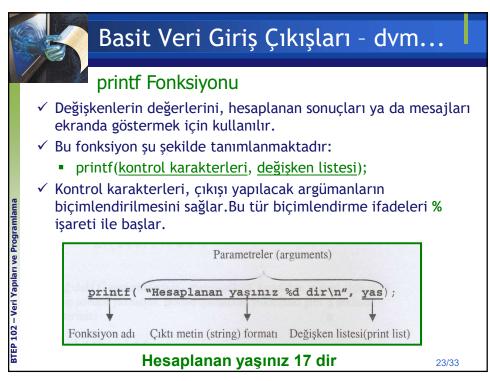
21

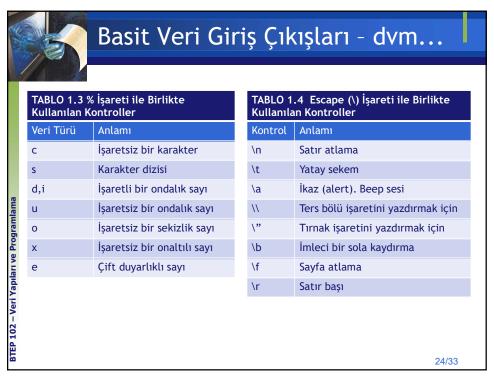


1.6 Basit Veri Giriş Çıkışları

- 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.

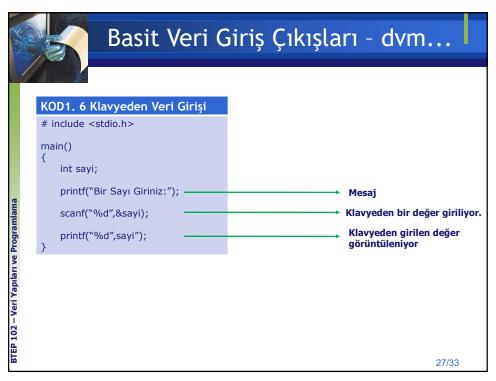
22/33







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. BTEP 102 – Veri Yapıları ve Programla Parametreler (arguments) "%d%f", &yas,&kilo); scanf (Değişken listesi(input list) Fonksiyon adı Girdi metin (string) formatı 26/33

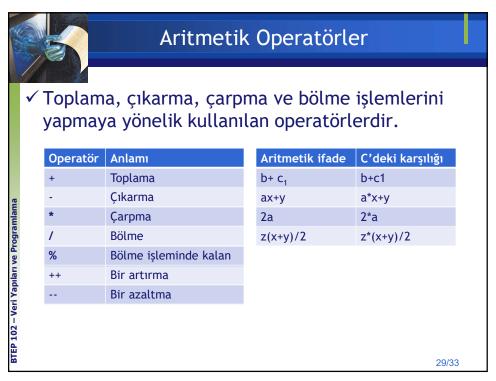




1.7 Operatörler (İşleçler)

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

28/33



```
Aritmetik Operatörler - devam...
                                     Kod 1.8 : ++ ve --
Operatörlerinin Kullanımı
       Kod 1.7: ++ ve --
                                                                     Kod 1.9: ++ ve --
       Operatörlerinin Kullanımı
                                                                     Operatörlerinin Kullanımı
                                                                     # include <stdio.h>
       # include <stdio.h>
                                     # include <stdio.h>
                                                                    main()
       main()
                                     main()
            int a=50;
                                          int a=50;
                                                                          int a=50;
            int b=30;
                                          int b=30;
                                                                          int b=30;
                                                                          int x,y;
           BTEP 102 – Veri Yapıları ve Programla
                                          a++;
                                          ++b;
                                                                          x=a++;
                                                                          y=++b;
                                         printf("%d\n",a );
                                                                         printf("%d\n",x);
printf("%d",y);
                                         printf("%d",b');
                                     Sonuç: 51
31
                                                                    Sonuç: 50
31
       Sonuç: 50
                                                                                             30/33
```



