BSM101 Programlama Dilleri I

Hafta 8
C Dilinde Değişken Tipleri ve Temel Giriş/Çıkış İşlemleri

Doç. Dr. Caner ÖZCAN

C Dilinde Temel Veri Tipleri

- C dilinde kullanılacak değişkenler ve veri tipleri programda önceden bildirilmek zorundadır.

Sayısal Veri Tipleri

a) Tamsayı Veri Tipleri

b) Ondalık Sayı Veri Tipleri

- int – tam sayılar
- float – virgüllü sayılar
- double – daha uzun ve çok hassas virgüllü sayılar
- char - karakterler

Tam Sayılar - Integer

- Tam sayıları ifade eder
  - Hem negatif hem pozitif tam sayılar
- C de tam sayıların (integer) ifade tarzı:

  `int`
- Örnek:

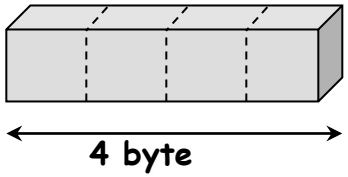
```
int toplam;    /* işaretli integer */

toplam = 100; /* pozitif olabilir */
toplam = -20; /* negatif olabilir */
```

```
int toplam = 32000; /* kodlama sırasında  */
                    /* ilk değer verilebilir */
```

Tam Sayılar – Integer (Devam)

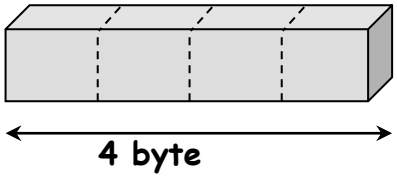
- Integer niteleyicileri: **long**, **short**, veya **unsigned**
- Integer değişkenlerin niteleyicilerine göre büyüklükleri değişir.
- Varsayılan integer büyüklüğü makine/işletim sistemine bağlıdır

int



-2.147.483.648 den 2.147.483.647 e kadar (toplam 4.294.967.296 adet sayı)

unsigned int



0 dan 4,294,967,295 e kadar (toplam 4,294,967,296 adet sayı)

Virgüllü Sayılar - float

- Gerçek sayıları ifade eder (virgüllü kısmıyla)
  - Pozitif ve negatif olabilir
- C de virgüllü sayıların ifade tarzı:

  float
- Örnek:

```
float f;

f = 0.12;     /* pozitif olabilir */
f = -245.56; /* negatif olabilir */
```

```
float f = 4.567; /* kodlama sırasında     */
                 /* ilk değer verilebilir */
```

Daha Uzun ve Çok Hassas Virgüllü Sayılar- double

- Standart "double precision floating point" (gerçek) sayılardır.
  - float gibi, fakat çok daha büyük ve hassastır.
- C deki ifade tarzı:

  double
- Örnek:

```
double d;

d = 3.12E+5;    /* 312000.0 */
d = -45.678;    /* negatif  */
```

```
double d = 4.567; /* ilk değer ataması */
```

Karakter - char

- Bir tek karakteri ifade eder
  - Karakterler
    - Alfabedeki büyük ve küçük harfler
    - 0 dan 9 a kadarki 10 numara
    - Özel semboller örneğin +#@½%&$.*?!£‘=-:/*^(){}[]~;,<>
- Karakterler tırnak işareti arasında kullanılır
  - örneğin. **'A'**
- C deki kullanım tarzı:

char

```
char c;

c = 'A';   /* A Harfi */
c = ‘9’;   /* 9 rakamı*/
```

```
char c = ‘c’; /* ilk değer verme */
```

Karakter (devam)

- Aslında karakterler 1 byte lık doğal sayıları ifade eder
  - char tipi değişkenler hafızada 1 byte yer tutar

- Karakterlerin (char değişkenleri) ASCII tablosundaki değerleri…
  - ‘A’ nın ASCII değeri 65
  - ‘B’ nın ASCII değeri 66

  - ‘0’ ın ASCII değeri 48
  - ‘1’ in ASCII değeri 49

  - http://www.asciitable.com/

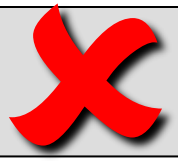
ASCII Tablosu

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Özel Karakterler

- Karakterler tek tırnak arasında gösterilir
  - Tırnak işaretini nasıl göstereceğiz?

```
char c;
c = ''';
```



```
char c;
c = '\'';
```

- Ters slaş(\) bir karakterin önünde kullanılırsa bu durum Escape Sequence (kaçış dizisi) olarak adlandırılır.
  - Kendisinden sonra gelecek Karakterin anlamını yok eder.

Escape Sequence – Kaçış Dizisi

- Ters slaş (\) ve bir karakterden oluşur. Derleyiciye sonraki karakterin normal olarak algılanması işaretini verir.
- Sık kullanılanlar
  - `\n` sonraki satıra geç
  - `\t` sonraki sekmeye geç
  - `\r` satır başına alır
  - `\\` ters slaş karakteri
  - `\'` tek tırnak
  - `\"` çift tırnak

Özet

- Tam sayılar (işaretli veya işaretsiz)
  - char – 1 byte
    - Aynı zamanda ASCII karakterleri depolamada kullanılır.
  - short – 2 byte
  - int – 4 byte
  - long – 4 veya 8 byte

- Gerçek sayılar(sadece işaretli)
  - float – 4 byte
  - double – 8 byte

Veri Tipleri ve Özellikleri

Veri tipi	Boyut	Aralık
char	1 byte	-128 : 127
unsigned char	1 byte	0 : 255
short	2 byte	-32768 : 32767
unsigned short	2 byte	0 : 65535
int	4 byte	-2147483648 : 2147483647
unsigned int	4 byte	0 : 4294967295
float (7 hassasiyet)	4 byte	1.175494e-38 : 3.402823e+38
double (16 hassasiyet)	8 byte	2.225074e-308 : 1.797693e+308

C Dilinin Temel Yazım Özellikleri

- Program yazımı belirli kalıpta, bloklar halinde olur.
- Bloklar, { } parantezleri ile oluşturulur.
- Komutlar aynı veya alt alta satırlara yazılabilir. Bir satıra en fazla 1023 karakter yazılabilir.
- Tüm komutlar, noktalı virgül (;) ile biter.
- Yalnız blok başlatan ifadelerden sonra noktalı virgül kullanılmaz.
- Programda kullanılan tüm değişkenler ve veri tipleri bildirilir.
- Programda kullanılacak olan komutların bulunduğu kütüphaneler aktifleştirilir/çağırılır.

C Dilinin Yapısı

- **Program başlığı:** Program hakkındaki açıklamaları içeren kısımdır.

  /* açıklamalar veya program başlığı */

- **Tanımlama ve Bildirme Bölümü:** Bu bölümde önişlemci komutları, değişken ve yapı tanımlamaları, sabit değer atamaları gibi bildirimler yer alır.

a) **include:** kütüphane çağırımı için kullanılır.

#include < *kütüphane adı* **>**

stdio.h: standart giriş/çıkış

conio.h: dos destekli giriş/çıkış

math.h: matematiksel fonksiyonlar

stdlib.h: dönüşüm, sıralama, arama vb.

C Dilinin Yapısı

- **Tanımlama ve Bildirme Bölümü:**

b) **define:** Bazı ifadelerin veya sabitlerin sembolik bir isme aktarılmasını sağlayan komuttur.

```
#define  sembolik_isim  eşdeğer_ifade
```

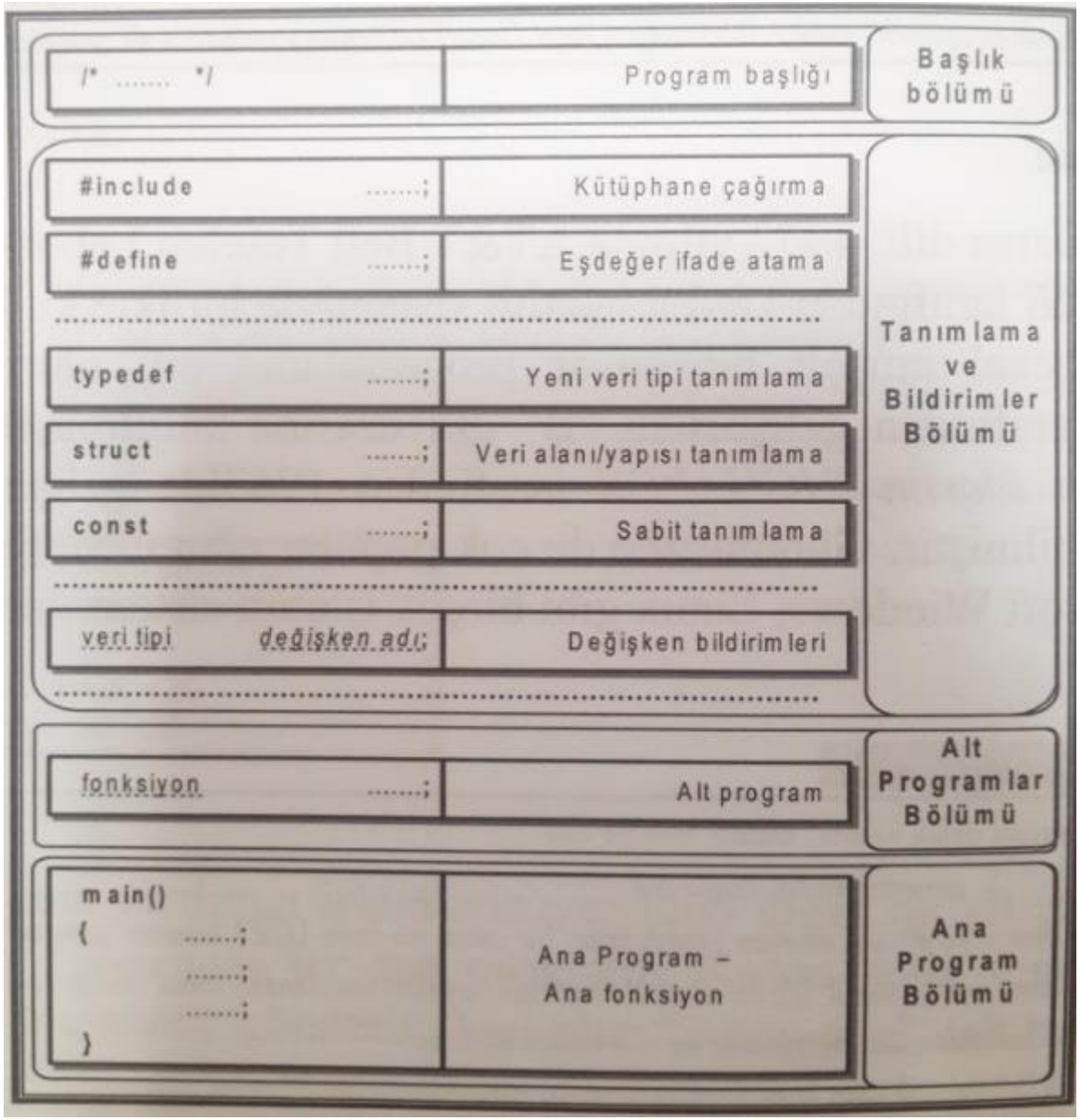
c) **Değişken bildirme:** C'de tüm değişkenler isim ve veri tipi olarak bildirilmektedir.

```
veri_tipi    değişken_adı;
veri_tipi    değişken_adı = değeri;
```

d) **Sabit tanımlama veya Başlangıç Değeri Verme:** C programlarında sabit tanımlamak için «const» kullanılmaktadır.

```
const  sabit_adı = değeri;
```

C Dilinin Yapısı



Structure of C Language

```c
/* Program:  Area Of Circle
   Author: Alien   */
#include<stdio.h>
#include<conio.h>

#define PI  3.14

void area(int);

main()
{
   int radius;
   printf("Enter Radius Of Circle ");
   scanf("%d",&radius);
   area(radius);
}

void area(int r)
{
   float result;
   result = PI*r*r;
   printf("Area Of Circle is %f", result);
}
```

Documentation Section

Link Section

Definition Section

Global Declaration Section

Main() Function Section

Subprogram Section

C Anahtar Kelimeleri

Anahtar Kelimeler (Keywords)			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Kod Yazımı İçin Tavsiyeler

- Program açıklamaları ve doküman hazırlama program yazıldıkça yapılmalıdır. Bu unutulmaması gereken çok önemli husustur.
- Değişken, sabit ve fonksiyon adları anlamlı kelimelerden seçilip yeterince uzun olmalıdır. Eğer bu isimler bir kaç kelimeden oluşacak ise, kelimeler alt çizgi (_) ile ayrılmalıdır veya her kelime büyük harfle başlamalıdır. Örneğin:
  - int son_alinan_bit;
  - void KesmeSayisi();
  - float OrtalamaDeger = 12.7786;
- Sabitlerin bütün harfleri büyük harfle yazılmalıdır. Örneğin:
  - #define PI = 3.14;
  - int STATUS 0x0379;

Kod Yazımı İçin Tavsiyeler

- Her alt yapıya girerken TAB tuşunu kullanın. Bu okunabilirliği arttıracaktır. Örneğin:

```
for(i=0;i<10;i++)
{
	for(j=0;j<i;j+=2)
	{
		do{
			k = i+j;
		}while(k!=0);
	}
}
```

Kod Yazımı İçin Tavsiyeler

- Aritmetik operatörler ve atama operatörlerinden önce ve sonra boşluk karakteri kullanın. Bu, yazılan matematiksel ifadelerin daha iyi anlaşılmasını sağlayacaktır.
- Örneğin:

```
Hmax = pow(Vo, 2) / (2 * g);
Tf = 2 * Vo / g;
Vy = Vo – g * t;
y = Vo * t - (g * t * t)/ 2.0;
z = ( a * cos(x) + b * sin(x) ) * acos(y);
```

Kod Yazımı İçin Tavsiyeler

- Program bittikten sonra tekrar tekrar programınızı inceleyerek, programınızı daha iyi şekilde yazma yollarını arayın.
- Aynı fonksiyonları daha kısa algoritmalarla ve daha modüler şekilde elde etmeye çalışın.
- Programınızın anlaşılması için gerekli çalışmaları yapın.
- Bilginizi ve eserinizi başkalarına en iyi şekilde aktarın.

Input/Output Kütüphanesi

- I/O fonksiyonları standart input/output C Kütüphanesinde tanımlanmış
  - stdio.h
- stdio.h ı programın başına eklemeniz gerekiyor
  - Bu eklemeyi #include önişlemci komutuyla yapmanız gerekiyor.

```
#include <stdio.h>
```

- Önişlemci komutları # ile başlar.
  - #define

Input/Output Fonksiyonları

- I/O fonksiyonları standart input/output C Kütüphanesinde tanımlanmış
  - stdio.h
- Klavye Input
  - scanf -- Genel Formatlanmış input
  - getchar -- tek bir karakter okur

- Monitör (Ekran) Output
  - printf -- Genel Formatlanmış output
  - putchar -- tek bir char (karakter) yazar

scanf Fonksiyonu

- Klavyeden belirtilen değişkene veri aktarılmasını sağlar.

scanf(" *biçim ifadesi* **", &**değişkenler listesi**);**

- Buradaki "biçim ifadesi" veri girişinin hangi biçimde olacağını; "değişkenler (adres) listesi" de verilerin aktarılacağı değişkenleri belirtir.

scanf Fonksiyonu

```
int number;

printf("Bir integer girin: ");

scanf("%d", &number);
```

Biçim kısmı

Değişken adresi

scanf Örnekleri

- “%c” char
- “%d” int
- “%f” float
- “%lf” double

```
int n;
double d;
char c;

printf(“3 deger giriniz;\n“);
printf(“bir int, bir double, ve bir char: “);
scanf(“%d”, &n);
scanf(“%lf”, &d);
scanf(“%c”, &c);
```

printf Fonksiyonu

- Ekrana veriyi biçimlendirerek yazabilen bir fonksiyondur.

```
printf( "biçim ifadesi", değişkenler);
```

- Çift tırnak arasında yer alan ‘biçim ifadesi’ genel olarak üç kısımdan oluşur.
  - Açıklama kısmı
  - Biçim kısmı
  - Kontrol/çıkış Kısmı

printf Fonksiyonu

```
int numara = 7;

printf("%d nin iki katı= %d \n", numara, 2*numara);
```

Biçim kısmı **İfade kısmı**

printf Fonksiyonu

a) **Açıklama:** Çift tırnaklar arasında verilip ekrana doğrudan yazılır.

```
printf("Ankara");
```

b) **Biçim:** % sembolüyle başlayan ve çıkış biçiminin belirlendiği kısımdır.

```
printf("Sonuc:  %d ", x);
```

.precision ⇨ maksimum kaç karakterde gösterileceğini belirtir.

```
printf("Sonuc: %.2lf ", y);
```

printf Tip Belirleme Karakterleri

Karakter	Tip	Çıkış Formatı
c	char	Tek byte karakter
hd	short	İşaretli ondalık short int (2 byte int)
d	int	İşaretli ondalık integer
ld	long	İşaretli ondalık long integer
u	int	İşaretsiz ondalık integer
x	int	Hexadecimal integer (16 tabanında)
f	float	İşaretli ondalıklı sayılar
lf	double	İşaretli ondalıklı sayılar fakat çok daha hassas
e	float double	İşaretli gerçek sayılar (bilimsel biçimlendirme)

printf Fonksiyonu

c) **Kontrol:** "\" işaretiyle başlayan bu karakterlerin anlamları şu şekildedir:

Karakter	**Anlamı**
\a	Ses üretir(alert)
\b	imleci bir sola kaydır(backspace)
\f	Sayfa atla. Bir sonraki sayfanın başına geç(formfeed)
\n	Bir alt satıra geç(newline)
\r	Satır başı yap(carriage return)
\t	Yatay TAB(Horizontal TAB)
\v	Dikey TAB(vertical TAB)
\"	Çift tırnak karakterini ekrana yaz
\'	Tek tırnak karakterini ekrana yaz
\\	\ karakterini ekrana yaz
%%	% karakterini ekrana yaz

printf Fonksiyon Örnekleri

```
double fp = 251.7366;
int i = 25;
printf("Reel sayi: %.2lf \n", fp);
printf(“Saga yaslanilmis integer: %10d \n”, i);
```

Çıktı:

```
Reel sayi: 251.74
Saga yaslanilmis integer :          25
```

printf Örnekler

```
printf("%.5f\n", 300.0123456789);
printf("%.14lf\n", 300.0123456789012345678 9);
```

```
300.01235
300.01234567890123
```

printf Örnekler

```
printf("%e ve %e\n",
       300.00145678901, 0.0024);
```

```
3.000015e+002 ve 2.400000e-003
```

float ve double için bilimsel görünüm.
Not: float için hassasiyet 7 rakamdır.

getchar ve putchar Fonksiyonları

- **getchar** klavyeden tek bir karakter alır.
- **putchar** ekrana tek bir karakter yazar.
- Örnek:

```
char c;

printf("Ne yapalım Menu \n");
printf("     (a) Bir C programı yazalım\n");
printf("     (b) Yüzmeye gidelim \n");
printf("     (c) TV izleyelim\n");
printf("Opsiyonlardan birini seç: ");

c = getchar(); /* Kullanıcı seçimini al*/
getchar();     /* yeni satıra geç '\n'*/
               /* karakterini koyar */
putchar('B');  /* Ekrana B yazdır */
c = 'Z';
putchar(c);    /* Ekrana Z yazdır */
```

Kaynaklar

- Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- Paul J. Deitel, “C How to Program”, Harvey Deitel.
- Bayram AKGÜL, C Programlama Ders notları