

## İfade Kavramı

x  
30  
x + 30  
x + 30 - y

Değişkenlerin, sabitlerin ve operatörlerin her bir bileşimine ifade denir.

## Nesne Kavramı

Bellekte yer kaplayan ve erişilebilir olan bölgelere nesne denir.

x bir nesnedir ve aynı zamanda bir ifadedir.

100 ifadedir, nesne belirtmez.

x + 10 bir ifadedir, nesne belirtmez.

nesne belirtir (lvalue)  $a = b + 25$  nesne belirtmez (rvalue)  
left value right value  
sol taraf değeri sağ taraf değeri

x → lvalue  
25 → rvalue  
b + 25 → rvalue

$a = x + y - 10;$   
a = 15;  
b = 20;  
 ~~$a + b = toplam;$   
 $a + 5 = b;$~~

## C Programlama Dilinde Veri Türleri

Tür Belirten Anahtar Sözcük	Uzunluk (byte) Windows (unix/linux)	Sınır Değerler
[signed] int	4 (4)	-2147483648, +2147483647
unsigned [int]	4 (4)	0, +4294967295
[signed] short [int]	2 (2)	-32768, +32767
unsigned short [int]	2 (2)	0, 65535
[signed] long [int]	4 (8)	-2147483648, +2147483647
unsigned long [int]	4 (8)	0, +4294967295
char	1 (1)	[-128, +127] [0, +255]
signed char	1 (1)	-128, +127
unsigned char	1 (1)	0, +255
<b>float</b>	4 (4)	$3,6 \cdot 10^{-38}$ , $3,6 \cdot 10^{+38}$
<b>double</b>	8 (8)	$1,8 \cdot 10^{-308}$ , $1,8 \cdot 10^{+308}$
<b>long double</b>	8 (8)	$1,8 \cdot 10^{-308}$ , $1,8 \cdot 10^{+308}$
[signed] long long	8 (8)	-9223372036854775808, +9223372036854775807
unsigned long long (C99 – C11)	8 (8)	0, 18446744073709551615
_Bool (C99 – C11)	1 (1)	true, false

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

```
int main()
```

```
{
```

```
    // <tür> degiskenAdi;
```

```
    int sayi;
```

```
    double a;
```

```
    printf("Bir sayi giriniz : ");
```

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

```
    printf("Girdiginiz sayi : %d\n", sayi);
```

```
return 0;
```

```
}
```

```
////////////////////////////////////  
////////////////////////////////////
```

## Derleyicilerin Hata Mesajları

- 1- Standartlarda belirtilen sentaks ve semantic kurallara uygun programları kesinlikle derlenmek zorundadır.
- 2- Standartlara uygun bir program başarılı olarak derlendikten sonra bir takım uyarı mesajları verebilir.
- 3- Sentaks ve semantic kurallara uyulmadığı her durumda derleyici bu durumu belirten bir mesaj vermek zorundadır. Bu mesaj error ya da warning olabilir.
- 4- Geçersiz bir program yine de başarılı olarak derlenebilir. Bir programın başarılı olarak derlenmesi onun geçerli olduğu anlamına gelmez.

## Bildirim ve Tanımlama Kavramları

C'de (C90) 3 yerde tanımlamak yapılır

- 1- Blok başlarında (yerel değişken)
- 2- Fonksiyon parameter parantezleri içerisinde
- 3- Tüm blokların başında. (global değişken)

```
#include <stdio.h>
int glb;    // global değişken
void foo() {
    glb = 10;
    printf("%d", glb);
}

int bar() {
    glb = 15 + 20;
    return glb;
}
int sum(int a, int b) { // parametre değişkenleri

    return a + b;
}
int main()
{
    // <tür> <degisken listesi>;
    /*
    int b;
    int c;
    int sayi;
    */
    int b, c, sayi;    // yerel değişken (local variable)
    double a;
    long l,
        k, m
        ;
    {
        float f;
        f = 3.20;
        {
            short sh;
```

```

        sh = 5;
    } // sh değişkeninin ömrü biter
} // f değişkeninin ömrü biter

printf("%d", glb);

return 0;
} //

/////////////////////////////////
/////////////////////////////////

```

#### Değişken Tanımlama Kuralları

- C programlama dili büyük-küçük harf duyarlılığı olan (case-sensitive) bir dildir.
- Değişkenler boşluk içermez.
 

```

sayi toplami >>> hatali kullanım
sayiToplami
sayi_toplami

```
- Değişken simlendirmede yalnızca ingilizce büyük ve küçük harfler, sayısal değerler ve alt çizgi kullanılır.
 

```

int sayi1;
int sayi1_toplami;

```
- Değişken isimleri her zaman bir harf ya da alt çizgi ile başlar.
 

```

int 1sayi; >> ERROR

```
- C standartlarına göre değişken uzunluklarında ilk 32 karakter dikkate alınır.
- Anahtar sözcükler değişken ismi olarak kullanılamaz.
 

```

if, for, while, goto, int

```

#### C Programlama Dilinde Kullanılan Anahtar Sözcükler

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

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

```

int main()
{
    int a = 10; // İlk değer verme (initialization)
    int b;
    b = 20; // ilk kez değer vermedir. ilk değer verme değildir.

    printf("a = %d, b = %d\n", a, b);
    printf("a = %d, b = %d\n", b, a);
    printf("%d%d\n", a, b);

    return 0;
}

/////////////////////////////////

```

////////////////////////////////////

## printf Format Karakterleri

Format Karakteri	Anlamı
%d	int, short ve char türlerini 10'luk sistemde yazmak için kullanılır
%ld	long int türünü 10'luk sistemde yazmak için kullanılır
%x, %X	int, short ve char türlerini hex sistemde yazmak için kullanılır
%lx, %lX	long int ve unsigned long int türlerini hex sistemde yazmak için kullanılır
%u	unsigned int, unsigned short ve unsigned char türlerini 10'luk sistemde yazmak için kullanılır
%lu	unsigned long int türünü 10'luk sistemde yazmak için kullanılır
%f	float ve double türlerini 10'luk sistemde yazmak için kullanılır
%lf	double türünü 10'luk sistemde yazmak için kullanılır
%c	char, short ve int türlerini karakter görüntüsü olarak yazar
%o	int, short ve char türlerini octal sistemde yazmak için kullanılır
%lo	long int ve unsigned long int türlerini octal sistemde yazmak için kullanılır

## Scanf Fonksiyonuyla Klavyeden Değer Okunması

```
#include <stdio.h>

int main()
{
    // klavyeden girilen sayilarin toplami
    int sayi1, sayi2, toplam;

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

    printf("ikinci sayiyi giriniz : ");
    scanf("%d", &sayi2);

    toplam = sayi1 + sayi2;

    printf("Toplam = %d\n", toplam);

    return 0;
}

////////////////////////////////////
////////////////////////////////////

#include <stdio.h>

int main()
{
    // klavyeden girilen sayilarin toplami
    double sayi1, sayi2, toplam;

    printf("Birinci sayiyi giriniz : ");
    scanf("%lf", &sayi1);

    printf("ikinci sayiyi giriniz : ");
    scanf("%lf", &sayi2);
```

```
    toplam = sayi1 + sayi2;

    printf("Toplam = %.2f\n", toplam);
    printf("Toplam = %lf\n", toplam);

    return 0;
}

////////////////////////////////////
////////////////////////////////////
```

**NOT:** double türü scanf fonksiyonuyla okunurken %lf kullanılmalıdır.