

## C DİLİNDE KONTROL İFADELER VE DÖNGÜLER

**If statement (if yapısı):** If yapısı, bir koşulun doğru olup olmadığını kontrol etmek için kullanılır. Eğer koşul doğruysa, if bloğu içindeki kod çalıştırılır. Aşağıdaki örnekte, koşul ( $x < 10$ ) doğru olduğu için, ekrana "x, 10'dan küçüktür" yazdırılır.

```
int x = 5;
if (x < 10) {
    printf("x, 10'dan küçüktür\n");
}
```

**Else if statement (else if yapısı):** Else if yapısı, if yapısından sonra kullanılır ve if bloğunda koşul doğru değilse çalıştırılır. Aşağıdaki örnekte, ilk koşul ( $x < 5$ ) yanlış olduğu için else if bloğu çalıştırılır ve ekrana "x, 5 ile 10 arasındadır" yazdırılır.

```
int x = 7;
if (x < 5) {
    printf("x, 5'ten küçüktür\n");
} else if (x < 10) {
    printf("x, 5 ile 10 arasındadır\n");
}
```

**Else statement (else yapısı):** Else yapısı, if ve else if bloğlarındaki koşulların hiçbirisi doğru değilse çalıştırılır. Aşağıdaki örnekte, ilk iki koşul yanlış olduğu için else bloğu çalıştırılır ve ekrana "x, 10'dan büyüktür" yazdırılır.

```
int x = 12;
if (x < 5) {
    printf("x, 5'ten küçüktür\n");
} else if (x < 10) {
    printf("x, 5 ile 10 arasındadır\n");
} else {
    printf("x, 10'dan büyüktür\n");
}
```

**While Loop (while döngüsü):** while ifadesi bir koşul doğru olduğu sürece bir döngü oluşturur. Örnek olarak, aşağıdaki kod parçasında, "i" değişkeni 0'dan başlayarak 10'a kadar artırılarak ekrana yazdırılır:

```
int i = 0;
while (i < 10) {
    printf("%d\n", i);
    i++;
}
```

**Do-While Statement( Do-while ifadesi):** Do-while ifadesi ilk olarak kod bloğunu çalıştırır ve sonra koşulu kontrol eder. Koşul doğru olduğu sürece döngü devam eder. Örnek olarak:

```
int i = 1;
do {
    printf("%d\n", i);
    i++;
} while (i <= 10);
```

Bu örnekte, i değişkeni 1'den 10'a kadar olan tüm sayıları ekrana yazdırır.

**Switch Case Statement( Switch case ifadesi):** Switch case ifadesi bir değişkenin belirli değerlerine göre çeşitli işlemler yapmak için kullanılan bir yapıdır. Örneğin, bir programda kullanıcı tarafından girilen bir sayının değerine göre ekrana farklı mesajlar yazdırmak isteyebiliriz. Bu durumda switch case ifadesi kullanılabilir.

Örnek kod aşağıdaki gibi olabilir:

```
int a = 2;
switch (a) {
    case 1: printf("Değer 1\n");    break;
    case 2: printf("Değer 2\n");    break;
    case 3: printf("Değer 3\n");    break;
    default:    printf("Değer 1, 2 veya 3 değil\n");
}
```

Bu örnekte, a değişkeninin değeri 2 olduğu için "Değer 2" yazdırılır.

**For Statement( For ifadesi):** For deyimi verilen şart doğru olduğu sürece belirtilen işlemlerin tekrarlanmasını sağlar. Aşağıda for döngüsünün söz dizimi görülmektedir.

```
for (başlangıç ; şart ; adım)
{
    komut
veya komutlar;
}
```

Örnek kod:

```
int i;
for ( i=0 ; i<10 ; i++)
{
    printf("i= %d\n",i);
}
```

Bu kod, ekrana "i = 0" ile "i = 9" arasındaki tüm tamsayıları yazdıracaktır. For döngüsünün üç bölümü vardır:

1. Başlangıç bölümü: Bu bölüm, döngü öncesi yapılacak işlemler için kullanılır. Burada, döngü sayacı olarak kullanılacak i değişkeni tanımlanır ve 0 değeri verilir.
2. Şart bölümü: Bu bölüm, döngünün devam etip etmeyeceğini belirler. Burada, i değişkeninin 10'dan küçük olduğu koşulu belirtilir. Eğer bu koşul sağlanıyorsa, döngü devam eder. Eğer koşul sağlanmıyorsa, döngü sona erer.

3. Adım bölümü: Bu bölüm, her döngü adımıda yapılacak işlemleri belirtir. Burada,  $i$  değişkeninin değeri 1 arttırılır.

For döngüsünün içinde, döngü sayacı olan  $i$  değişkeni kullanılabilir. Bu örnekte,  $i$  değişkeninin değeri her döngü adımıda 1 arttırılır ve ekrana yazdırılır. Bu döngü,  $i$  değişkeninin değerinin 10'dan küçük olduğu sürece devam eder.

