

BÖLÜM 6: KARŞILAŞTIRMALI KONTROL YAPILARI

C programlama dilinde, diğer programlama dillerinde olduğu gibi, işlemler, ilk satırdan başlamak üzere sırayla çalışır. Program kontrol yapıları ise, programın yukarıdan aşağıya doğru olan bu akışını değiştirir. Programın akışı bir veya birkaç koşulun gerçekleşmesine bağlı olarak değiştirilebileceği gibi, herhangi bir koşulun gerçekleşmesi aranmaksızın da değiştirilebilir.

C programlama dilinde, program kontrol yapıları iki ayrı kategoride toplanabilir. Birinci kategoride `if` ve `switch` gibi belirli bir koşula bağlı olarak programın akışını değiştiren kontrol yapıları vardır. İkinci kategoride ise `for` ve `while` gibi, yine belirli bir koşulun gerçekleştiği sürece bir işlem veya işlem bloğunu tekrar tekrar çalıştıran kontrol yapıları vardır. Bu çeşit yapılara döngü (loop) kontrol yapıları da denir.

Bu bölümde, koşula bağlı olarak programın akış yönünü değiştiren yapılar, daha sonraki bölümde ise döngü (loop) kontrol yapıları incelenecektir.

I. `if` KONTROL YAPISI

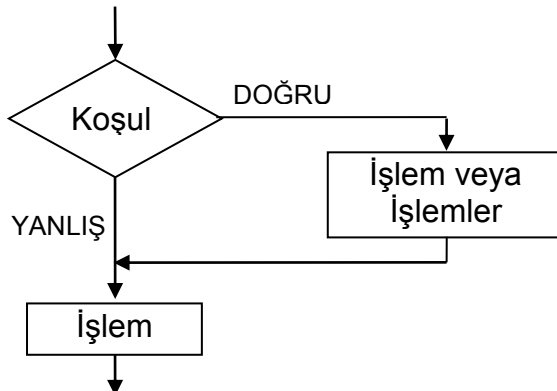
`if` kontrol yapısı, bir veya birden fazla koşulun gerçekleşmesine bağlı olarak bir işlem veya işlem bloğunun çalışmasını sağlar. `if` yapısının formatı aşağıdaki gibidir:

```
if (koşul)
    işlem;
```

Koşul, bir değişken, ifade veya sabit olabilir. Koşul parantez içinde yazılır. Eğer, koşula bağlı olarak çalıştırılması istenen işlemler birden fazla ise, bu işlemler küme parantezi (`{ }`) içine alınır. Bloktaki işlemler, birbirinden noktalı virgül (`;`) ile ayrılır.

```
if (koşul)
{
    işlem 1;
    işlem 2;
    .
    işlem n;
}
```

Aşağıdaki şekilde, `if` kontrol yapısının akış diyagramı görülmektedir.



Program kontrolü, `if` kontrol yapısıyla karşılaştığında, parantez içinde verilen koşulun doğru olup olmadığını kontrol eder. Koşul DOĞRU ise bu koşuldan sonra gelen işlem veya işlemleri icra eder. Koşul YANLIŞ ise, koşul cümlesinden sonra gelen işlem veya işlemler bloğu çalıştırılmaz, bloktan bir sonraki işleme geçilir.

Aşağıdaki programda, klavyeden girilen sayının 100'a eşit, 100'dan küçük veya 100'dan büyük olup olmadığı kontrol edilmekte ve buna göre akış yönü değiştirilmektedir.

```
#include <stdio.h>
void main()
{
    int a;
    printf ("Lütfen bir sayı giriniz: ");
    scanf ("%d", &a);
    if (a == 100)
        printf("100 sayısını girdiniz.\n");
    if (a < 100)
        printf("100 den küçük bir sayı girdiniz.\n");
    if (a > 100)
        printf("100 den büyük bir sayı girdiniz.\n");
}
```

Program çıktısı:

```
Lütfen bir sayı giriniz: 12
100 den küçük bir sayı girdiniz.
```

Programın akışı, birden fazla koşulun gerçekleşmesi durumunda değiştirilmek isteniyorsa bu koşullar `&&` (VE) bağlacıyla bağlanır. Aşağıdaki programda, iki koşulun birlikte gerçekleşmesi durumunda programın akışı değiştirilmektedir.

```
/* VE bağlacının kullanılması */
#include <stdio.h>
void main()
{
    int a;
    printf ("Lütfen pozitif bir tam sayı giriniz: ");
    scanf ("%d", &a);
    if(a > 0 && a <= 10)
        printf("1 ile 10 arasında bir sayı girdiniz.\n");
    if(a > 10 && a <= 100)
        printf("11 ile 100 arasında bir sayı girdiniz.\n");
    if(a > 100)
        printf("100 den büyük bir sayı girdiniz.\n");
}
```

Program çıktısı:

```
Lütfen pozitif bir tam sayı giriniz: 30
11 ile 100 arasında bir sayı girdiniz.
```

Yukarıdaki program çalıştırıldığında, klavyeden girilen 30 sayısı `a` değişkenine aktarılır. Birinci `if` cümlesinde, `a` değişkeninin tuttuğu sayının sıfırdan büyük VE 10'a eşit ya da küçük olup olmadığı kontrol ediliyor, `a` değişkeninin içindeki 30

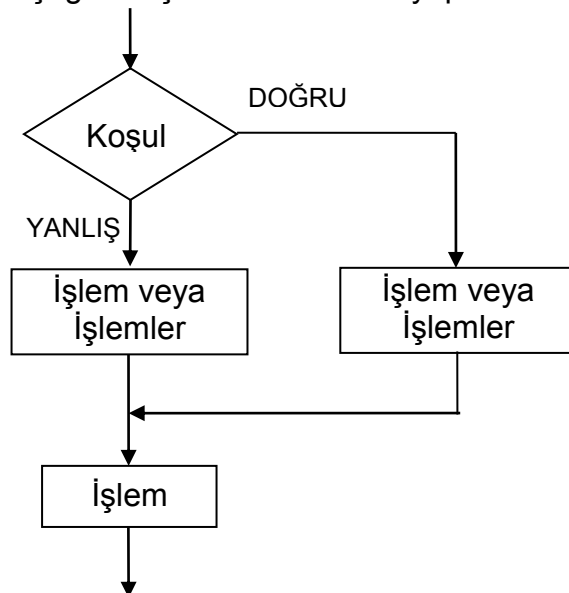
sayısı bu iki koşulu taşımadığı için, birinci koşul cümlesinden sonra yazılmış olan işlem çalıştırılmaz ve kontrol bir alttaki `if` cümlesine geçer. İkinci `if` cümlesinde, `a` değişkeninin tuttuğu sayının 10'dan büyük VE 100'e eşit ya da küçük olup olmadığı kontrol edilmektedir. `a` değişkeninin tuttuğu 30 sayısı, bu iki koşuluda taşıdığı için, ikinci `if` cümlesinden sonra yazılmış olan `printf` fonksiyonu çalıştırılacaktır. Üçüncü `if` cümlesinde ise `a` değişkeninin tuttuğu sayının 100'den büyük olup olmadığını kontrol etmektedir. `a` değişkeninin tuttuğu sayı 30 olduğu için bu `if` cümlesinden sonra gelen `printf` fonksiyonu çalıştırılmayacaktır.

II. `if-else` KONTROL YAPISI

`if-else` yapısının formatı aşağıdaki gibidir:

```
if (koşul)
    işlem; veya
    {
        işlem 1;
        işlem 2;
        .
        işlem n;
    }
else
    işlem; veya
    {
        işlem 1;
        işlem 2;
        .
        işlem n;
    }
```

Parantez içinde verilen koşul DOĞRU ise parantezden sonraki işlem veya işlemler bloğu çalıştırılır, `else` sözcüğünden sonraki işlem veya işlemler bloğu çalıştırılmaz. Koşul YANLIŞ ise parantezden sonraki işlem veya işlemler bloğu çalışmaz, `else` sözcüğünden sonraki işlem veya işlemler bloğu çalıştırılır. Aşağıdaki şekilde `if-else` yapısının akış diyagramı görülmektedir.



Aşağıdaki programda if-else yapısı kullanılmıştır.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char ch;
    printf ("Bir harf tuşuna basınız: ");
    ch=getche();
    printf("\n");
    if(ch >= 'A' && ch <= 'Z')
    {
        printf("Büyük harf tuşuna bastınız.\n");
        printf("Bastığınız tuş: %c\n", ch);
    }
    else
    {
        printf("Bastığınız tuş büyük harf tuşu değil.\n");
        printf("Bastığınız tuş: %c\n", ch);
    }
}
```

Program çıkışı:

```
Bir harf tuşuna basınız: b
Bastığınız tuş büyük harf tuşu değil.
Bastığınız tuş: b
```

Program çalıştırıldığında klavyeden bir tuşa basılması istenecektir. b tuşuna bastığımızı kabul edelim. if kelimesinden sonra parantez içinde yazılmış olan koşul cümlesinde, girilen karakterin A ile Z arasında yer alıp almadığı kontrol edilmektedir. Girilen harf b olduğuna göre, bu koşul gerçekleşmemiş demektir, çünkü b küçük harf olduğu için A ile Z arasında yer almaz. Parantez içinde verilen koşul YANLIŞ olduğu için, parantezi takip eden işlemler yapılmaz, else sözcüğünü takip eden işlem yapılır.

Aşağıdaki programda da if-else koşullu dallanma yapısı kullanılmıştır.

```
#include <stdio.h>
void main()
{
    int a;
    printf ("Lütfen bir sayı giriniz: ");
    scanf("%d",&a);
    if(a == 10)
        printf("10 sayısını girdiniz.\n");
    else
        if(a < 10)
            printf("10 dan küçük bir sayı girdiniz.\n");
        else
            printf("10 dan büyük bir sayı girdiniz.\n");
}
```

Program çıktısı:

```
Lütfen bir sayı giriniz: 12
10 dan büyük bir sayı girdiniz.
```

Yukarıdaki program çalıştırıldığında, klavyeden 12 sayısını girdiğimizi kabul edelim. `scanf` fonksiyonu, klavyeden girilen 12 sayısını `a` değişkenine aktaracaktır. Programdaki ilk `if` sözcüğünden sonra parantez içinde (`a==10`) yazılarak, `a` değişkeninin tuttuğu sayının 10 sayısına eşit olup olmadığı kontrol edilmektedir. Bu durum (yani 12 sayısının 10 sayısına eşit olma durumu) doğru olmadığı için parantezi takip eden işlem gerçekleşmez ve kontrol `else` sözcüğüne geçer. `else` ten sonra parantez içersinde (`a<10`) koşulunun doğru olup olmadığı kontrol edilmektedir. `a` değişkeninde, klavyeden girilen 12 sayısının bulunduğunu biliyoruz. O halde (`a<10`) ifadesi de yanlıştır. Bu nedenle, parantezi takip eden işlem değil bu bloğun `else` sözcüğünü takip eden işlem icra edilecektir.

Birçok program, MENÜ sistemine göre çalışmaktadır. MENÜ sisteminde, programı kullanan operatöre çeşitli seçenekler sunulmakta, operatör yapmak istediği işlemi seçmekte ve seçilen bu işlemi icra eden program parçası çalışmaktadır.

```
#include <stdio.h>
void main()
{
    int a, b, c, i;
    printf("Birinci sayıyı giriniz: ");
    scanf("%d", &a);
    printf("İkinci sayıyı giriniz: ");
    scanf("%d", &b);
    printf("\n\n");
    printf("MENÜ\n");
    printf("=====\n");
    printf("1-Toplama\n");
    printf("2-Çıkarma\n");
    printf("3-Çarpma\n");
    printf("\n");
    printf("Tercihinizi giriniz (1/2/3): ");
    scanf("%d", &i);
    printf("\n");
    if(i==1)
    {
        c=a+b;
        printf("Toplam :%d\n", c);
    }
    else if(i==2)
    {
        c=a-b;
        printf("Fark :%d\n", c);
    }
    else if(i==3)
    {
        c=a*b;
        printf("Çarpım :%d\n", c);
    }
    else
        printf("1, 2 veya 3 girmelisiniz.\n");
}
```

Program çıktısı:

```
Birinci sayıyı giriniz: 8
İkinci sayıyı giriniz: 5
```

```
MENÜ
=====
1-Toplama
2-Çıkarma
3-Çarpma
```

```
Tercihinizi giriniz (1/2/3): 2
Fark: 3
```

Bu program çalıştırıldığında klavyeden iki sayı girilmesi istenir. Daha sonra ekranda bir MENÜ görüntülenir. Menü seçenekleri Toplama, Çıkarma ve Çarpma işlemleridir ve her işlemin başında bir rakam vardır. Operatöre tercihinin ne olduğu sorulur. Klavyeden 1 girilirse toplama, 2 girilirse çıkarma, 3 girilirse de çarpma işlemi yapılır. Bu sayıların dışında bir değer girilirse “1, 2, 3 dışında bir değer girdiniz.” şeklinde ekrana mesaj yazılacaktır.

III. switch-case KONTROL YAPISI

switch-case kontrol yapısında bir değişken veya ifade, birkaç ayrı değerle karşılaştırılmakta; hangisine eşit ise program o kola yönelmektedir. switch-case yapısının formatı aşağıdaki gibidir.

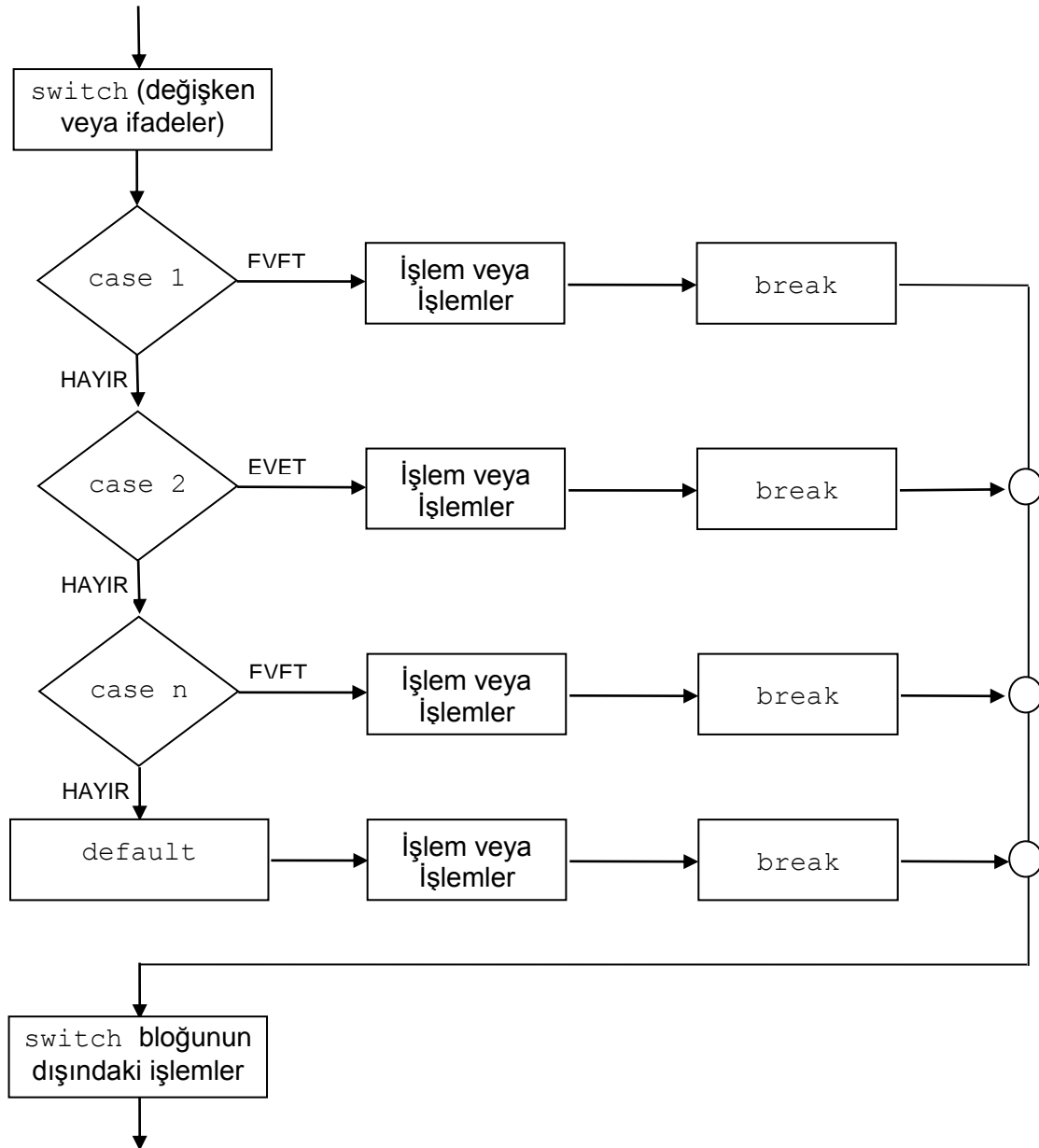
```
switch (değişken veya ifade)
{
    case sabit1:
        işlem1;
        işlem2;
        . . .
        işlemn;
        break;
    case sabit2:
        işlem1;
        işlem2;
        . . .
        işlemn;
        break;
    . . .
    case sabitn:
        işlem1;
        işlem2;
        . . .
        işlemn;
        break;
    default:
        işlem1;
        işlem2;
        . . .
        işlemn;
        break;
}
```

switch sözcüğünden sonra parantez içinde verilen değişken veya ifade char veya integer tipi bir değer olabilir, karakter stringi veya float tipi sayı olamaz. Aynı şekilde, case kelimesinden sonra da integer veya char tipi bir sabit kullanılır. char tipi bir sabit kullanılıyorsa, bu sabit, ' ' (tek tırnak) karakteri içine alınmalıdır.

default, program akışının üstteki case'lerden hiçbirine saptmaması halinde yapılacak olan işlem veya işlemleri belirtir.

case ve default sözcükleriyle başlayan işlem bloklarının sonunda break sözcüğü yer alır. break, ilgili blok işlem gördükten sonra, switch ile başlayan bloğun dışına çıkılmasını sağlar.

Aşağıdaki şekilde, switch-case kontrol yapısının akış diyagramı görülmektedir.



Aşağıdaki programda switch-case yapısı kullanılmıştır.

```
#include <stdio.h>
void main()
{   int a;
    printf("1, 2 veya 3 sayısını giriniz: ");
    scanf("%d",&a);

    switch(a)
    {   case 1:
        printf("BİR sayısını girdiniz...\n");
        break;
        case 2:
        printf("İKİ sayısını girdiniz...\n");
        break;
        case 3:
        printf("ÜÇ sayısını girdiniz...\n");
        break;
        default:
        printf("1,2,3 dışında bir sayı girdiniz...\n");
        break;
    }
}
```

Program çıktısı:

```
1, 2 veya 3 sayısını giriniz: 2
İKİ sayısını girdiniz...
```

Bu program çalıştırıldığında, klavyeden girilen sayıya göre program kontrolü dallanır. Klavyeden 2 sayısı girildiğinde, kontrol case 2: satırına aktarılır ve buradaki işlem yapılır. Bu programda default kullanıldığı için 1, 2 ve 3 dışında bir sayı girildiğinde, kontrol default bloğuna aktarılır ve ekrana 1,2,3 dışında bir sayı girdiniz... yazılır ve switch bloğundan çıkılır. Eğer default kullanılmamış olsaydı 1, 2 ve 3 dışında bir sayı girildiğinde, kontrol hiçbir bloğa aktarılmadan switch bloğundan çıkacaktı.

Aşağıdaki programda ise klavyeden girilen alfabetik karaktere göre kontrol yapılmaktadır.

```
#include <stdio.h>
void main()
{   char x;
    printf("a, b veya c tuşuna basınız: ");
    x=getchar();

    switch(x)
    {   case 'a':
        printf("AHMET...\n");
        break;
        case 'b':
        printf("BURAK...\n");
        break;
    }
```



```

        case 'c':
            printf("CEM...\n");
            break;
        default:
            printf("a, b ve c dışında bir harf girdiniz.\n");
            break;
    }
}

```

Program çıktısı:

a, b veya c tuşuna basınız: b
BURAK...

Daha önce if yapısı kullanılarak bir MENÜ programı yazılmıştı. Şimdi de aynı işlemleri switch-case yapısı kullanarak yapan programı inceleyelim.

```

#include <stdio.h>
void main()
{
    int a, b, c, i;
    printf("Birinci sayıyı giriniz: ");
    scanf("%d", &a);
    printf("İkinci sayıyı giriniz: ");
    scanf("%d", &b);
    printf("\n\n");
    printf("MENÜ\n");
    printf("=====\n");
    printf("1-Toplama\n");
    printf("2-Çıkarma\n");
    printf("3-Çarpma\n");
    printf("\n");
    printf("Tercihinizi giriniz (1/2/3): ");
    scanf("%d", &i);
    printf("\n");
    switch(i)
    {
        case 1:
            c=a+b;
            printf("Toplam :%d\n",c);
            break;
        case 2:
            c=a-b;
            printf("Fark :%d\n",c);
            break;
        case 3:
            c=a*b;
            printf("Çarpım :%d\n",c);
            break;
        default:
            printf("1, 2, 3 dışında bir sayı girdiniz...\n");
            printf("Programı yeniden çalıştırınız...\n");
            break;
    }
}

```

Program çıktısı:

Birinci sayıyı giriniz: 5
İkinci sayıyı giriniz: 3

MENÜ

=====

1-Toplama
2-Çıkarma
3-Çarpma

Tercihinizi giriniz (1/2/3): 3
Çarpım: 15

Aşağıdaki programda ise `switch-case` yapısının değişik bir kullanımı görülüyor. Bu program, klavyeden girilen harflerin büyük, küçük, sesli veya sessiz harf olduğunu bulmaktadır.

```
#include <stdio.h>
void main()
{
    char a;
    printf("Bir harf giriniz: ");
    a=getchar();

    switch(a)
    {
        case 'A':
        case 'E':
        case 'I':
        case 'U':
        case 'O':
            printf("büyük, ");
            printf("sesli harf...\n");
            break;
        case 'a':
        case 'e':
        case 'i':
        case 'u':
        case 'o':
            printf("küçük, ");
            printf("sesli harf...\n");
            break;
        default:
            printf("sessiz harf...\n");
            break;
    }
}
```

Program çıktısı:

Bir harf giriniz: E
büyük, sesli harf...

IV. ? KARŞILAŞTIRMA YAPISI

C dilinde `if-else` karşılaştırma deyiminin yaptığı işi sınırlı olarak yapan bir operatördür. Genel yazım biçimi:

```
(koşul) ? deyim1 : deyim2;
```

İlk önce koşul sınanır. Eğer koşul doğruysa deyim1 aksi takdirde deyim2 değerlendirilir. deyim1 ve deyim2 de atama işlemi yapılamaz. Ancak koşul deyiminde atama işlemi yapılabilir. deyim1 ve deyim2 yerine fonksiyon da kullanılabilir. Aşağıda bu deyimmin kullanımına ait örnekler verilmiştir.

```
x = ( a > b ) ? a : b;
```

Yukarıdaki ifadede koşul a'nın b'den büyük olmasıdır. Eğer koşul doğruysa x adlı değişkene a, değilse b değeri atanır. Bu şekilde kullanım if-else yapısı ile kurulmak istenirse:

```
if( a > b ) x = a;
    else x = b;
```

şeklinde olacaktır.

? ve if-else yapısının kullanımını gösteren program.

```
#include <stdio.h>
void main()
{
    int x,y,z;
    printf("x : ");
    scanf("%d",&x);      /* x okunuyor */
    printf("y : ");
    scanf("%d",&y);      /* y okunuyor */
    if( x )               /* x 0 dan farklı mı? */
        z = ( y > x ) ? x*y : x+y;
                        /* y>x ise z=x*y, değilse z=x+y */
    else
        z = 0;
    printf("z = %d\n",z);
}
```

Örnek program çıktıları:

x : 0	x : 5	x : 8
y : 6	y : 9	y : 2
z = 0	z = 45	z = 10

Tamsayı olarak tanımlanan x, y, z değişkenlerinden x ve y ye değer okutulmuştur. if deyimindeki koşul biraz farklıdır. Genel olarak koşul bu şekilde bildirilirse, koşulun 0'dan farklı olup olmadığı sınanır. Yani if(x) ile if(x!=0) aynı anlamdadır. Bu kullanım çok yaygındır. Eğer x 0'dan farklı ise koşul olumlu olarak değerlendirilecektir. ? ile bir sına yapılmaktadır. Eğer y, x den büyük ise z değişkenine x*y, aksi takdirde x+y değeri atanmaktadır. Eğer x=0 ise z değişkenine 0 değeri atanmaktadır. if ve else deyimlerinden sonra sadece tek bir işlem yapılacağından '{' ve '}' karakterlerinin kullanılmadığına dikkat edin.