



BİLGİSAYAR PROGRAMLAMA 1

Ders Notu 5– Karşılaştırma İfadeleri

Konya Teknik Üniversitesi
Elektrik – Elektronik Mühendisliği Bölümü

15.03.2024

Konya

Koşul (Şart) İfadeleri

- Koşul ifadelerinin ve akış kontrollerinin programlama dillerinde ve uygulama altyapısında her zaman önemli bir yeri vardır.
- Yazdığımız uygulamanın hangi koşulda nasıl davranacağını belirterek bir akış kontrolü oluşturabilmek için koşul ifadelerinin iyi bilinmesi gereklidir.
- Tüm programlama dillerinde olduğu gibi en temel koşul ifadesi «if-else» bloklarıdır.

Koşul (Şart) İfadeleri

- Sonucu DOĞRU veya YANLIŞ olan herhangi bir ifadeye “koşul” denir.
- DOĞRU => 1 ile gösterilir.
- YANLIŞ => 0 ile gösterilir.
- Örnek:

```
if(a==b) {  
    printf("İki sayı birbirine eşittir \n");  
}
```

- **(a==b)** ifadesi bir koşuldur.
- Koşullar karşılaştırma operatörleri ve mantıksal operatörler kullanılarak oluşturulurlar.

If - Else

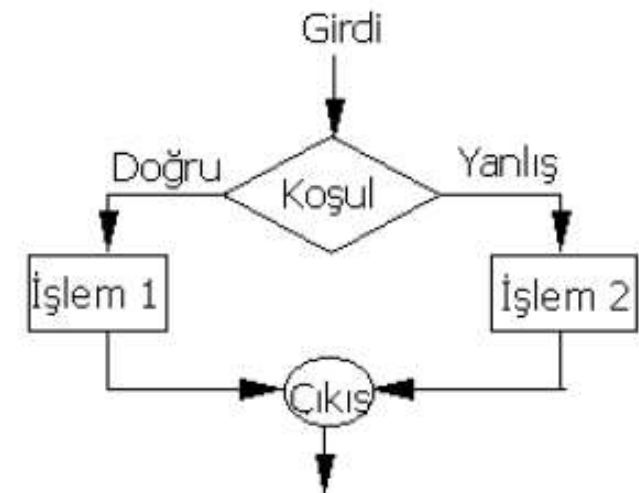
- **if** ve **else** tek bir karşılaştırma deyimi olup **else** kullanımı isteğe bağlıdır.
- Eğer bu koşul olumlu ise **if** den sonraki bölüm yürütülür ve **else** den sonraki bölüm atlanır.
- Koşul olumsuz ise **if** den sonraki küme atlanır ve eğer varsa, **else** den sonraki kümedeki işlemler gerçekleştirilir.

if (koşul)

{ Doğru ise İşlem1 }

else

{ Yanlış ise İşlem2 }



If - Else

Örnek 1

```
if (sayi < 0) {  
    printf("Negatif sayı girdiniz\n");  
}  
else {  
    printf("Sıfır veya pozitif bir sayı  
girdiniz\n");  
}
```

Örnek 2

```
if (sayi < 0)  
    printf("Negatif sayı girdiniz\n");  
else  
    printf("Sıfır veya pozitif bir sayı  
girdiniz\n");
```

Not: Her iki örnek aynı işlemi gerçekleştirmektedir. Tek fark birinci örnekte { } sembollerinin kullanılmasıdır. Tek satır deyim içeren bloklarda, { } sembollerinin kullanılmasına gerek yoktur.

İç içe if – else blokları

- **if-else** yapıları içerisine başka **if-else** yapıları yerleştirerek birden fazla koşul aynı anda test edilebilir.
- **Örnek:**

```
if (sayi < 0)
    printf("Negatif sayı girdiniz\n");
else
    if (sayi > 0)
        printf("Pozitif bir sayı girdiniz\n");
    else
        printf("Sıfır sayısını girdiniz\n");
```

Peşpeşe if – else blokları

- 2’den fazla durum ihtimali varsa:

```
if(koşul_1) {  
    deyimler; //(küme_1)  
}  
else if(koşul_2) {  
    deyimler; //(küme_2)  
}  
....  
....  
else if(koşul_n-1) {  
    deyimler; //(küme_n-1)  
}  
else {  
    deyimler; //(küme_n)  
}
```


Girilen Notun Harf Karşılığını Bulma

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int ogrenci1;
6      printf ("Notu giriniz:\n");
7      scanf ("%d",&ogrenci1);
8
9      if(ogrenci1<=24)
10         printf ("Ogrencinin Harf Notu FF");
11     else if(ogrenci1>=25 && ogrenci1<=34)
12         printf ("Ogrencinin Harf Notu FD");
13     else if(ogrenci1>=35 && ogrenci1<=39)
14         printf ("Ogrencinin Harf Notu DD");
15     else if(ogrenci1>=40 && ogrenci1<=49)
16         printf ("Ogrencinin Harf Notu DC");
17     else if(ogrenci1>=50 && ogrenci1<=57)
18         printf ("Ogrencinin Harf Notu CC");
19     else if(ogrenci1>=58 && ogrenci1<=64)
20         printf ("Ogrencinin Harf Notu CB");
21     else if(ogrenci1>=65 && ogrenci1<=73)
22         printf ("Ogrencinin Harf Notu BB");
23     else if(ogrenci1>=74 && ogrenci1<=81)
24         printf ("Ogrencinin Harf Notu BA");
25     else if(ogrenci1>=82 && ogrenci1<=100)
26         printf ("Ogrencinin Harf Notu AA");
27     else
28         printf ("Uygun Deger Girmediniz");
29     return 0;
30 }
```

<u>Not Aralıkları</u>	<u>Harf Notu</u>
82 – 100	AA
74 – 81	BA
65 – 73	BB
58 – 64	CB
50 – 57	CC
40 – 49	DC
35 – 39	DD
25 – 34	FD
0 – 24	FF

İpucu (if – else)

- C programlamada, koşulların mantıksal değerlendirmelerinde, **beklenen değere ulaşıldığında** koşulun tamamı değerlendirilmeden koşuldan çıkılır.
- **VEYA** kullanımında koşulun ilk kısmı **DOĞRU** (**true**) ise koşulun tamamı **DOĞRU** olacağı için koşulun ikinci kısmı değerlendirilmez.
- **VE** kullanımında koşulun ilk kısmı **YANLIŞ** (**false**) ise koşulun tamamı **YANLIŞ** olacağı için koşulun ikinci kısmı değerlendirilmez.
- **Örnek:** **if** ((a>b) **&&** (a<c)) eğer (a>b) **YANLIŞ** ise (a<c) değerlendirilmez.
- **Örnek:** **if** ((a>b) **||** (a<c)) eğer (a>b) **DOĞRU** ise (a<c) değerlendirilmez.

Örnek

1. Program Parçası

```
if (x < 5)
if (y > 5)
printf("aaaa\n");
else
printf("bbbb\n");
printf("cccc\n");
```

Soru 1: x=4 ve y=6 iken
çıktı ne olur?

aaaa
cccc

Soru 2: x=6 ve y=4 iken
çıktı ne olur?

cccc

2. Program Parçası

```
if (x < 5) {
if (y > 5)
printf("aaaa\n");
}
else {
printf("bbbb\n");
printf("cccc\n");
}
```

aaaa

bbbb
cccc

If – Else Kullanım Hataları

Hata-1

```
if sayi==10 //Söz dizimi hatası
    printf ("Girilen sayi 10\n");
```

- Koşul ifadesinde () sembolleri kullanılmadığı için “**syntax error**” hatası alınır (Söz dizimi hatası).

Hata-2

```
if (sayi==10); //Boş ifade
    printf ("Girilen sayi 10\n");
```

- Koşul ifadesinden sonra ; sembolü kullanılmıştır. C derleyicisi için boş bir ifadedir. Sonraki satır bu satırdan bağımsızdır.

If – Else Kullanım Hataları

Hata-3

```
if (10<=sayi<=50) //Mantıksal Hata
```

- Sonucu **her zaman doğru** olan mantıksal bir hatadır.
- Koşul ifadesi soldan sağa doğru değerlendirilir. **sayi** değişkenin değeri 10 ile karşılaştırılır.
 - 10'dan küçükse **false** yani **0** sonucu döner,
 - 10'dan büyükse **true** yani **1** sonucu döner,
 - Her iki durum da (0 veya 1 sonucu) dönen değer 50'den küçük olacağı için bu ifade her zaman doğru sonuç verir.
- Bunun yerine `((sayi>=10) && (sayi<=50))` koşul ifadesi kullanılmalıdır.

If – Else Kullanım Hataları

Hata 4

```
if (sayi=10) //Mantıksal Hata
```

- Sonucu **her zaman doğru** olan mantıksal bir hatadır.
 - Koşul ifadesinde karşılaştırma operatörü `==` yerine, atama operatörü `=` kullanılmıştır.
 - Atama sonucunda 10 değeri oluşur.
 - 10 değeri 0'dan büyük olduğu için sonuç “true” dönecektir ve bu da yanlış bir sonuç üretecektir.
- Atama operatörü yerine, `==` karşılaştırma operatörü kullanılmalıdır.

Switch – Case Yapısı

- Bir **değişkenin** aldığı **değerlere** bağlı olarak, program akışındaki belirli işlemlerin yaptırılmasını sağlar.
- Aslında bu tür işlemleri **if** deyimi yardımıyla yerine getirebiliriz.
- Ancak bazı uygulamalarda **if** deyiminin kullanılması programın **karmaşıklığını** artırabilir ve **okunabilirliğini** azaltabilir.
- Eğer bir değişkenin değeri belirli **sabitlerle** karşılaştırılacak ve bunun sonucunda farklı işlemler yapılacak ise **if** deyimi yerine **switch** deyimi kullanmamız daha sağlıklı olacaktır.
- Sadece **char** ve **int** veri türleri için kullanılabilir.

Switch – Case Yapısı

```
switch (değişken) {
```

```
    case sabit1:  
        deyim1  
        break;
```

değişken değeri **sabit**₁ değerine eşitse **deyim**₁ çalıştırılır ve **break** kullanılarak bloktan çıkış yapılır.

```
    case sabit2:  
        deyim2  
        break;
```

```
    ...
```

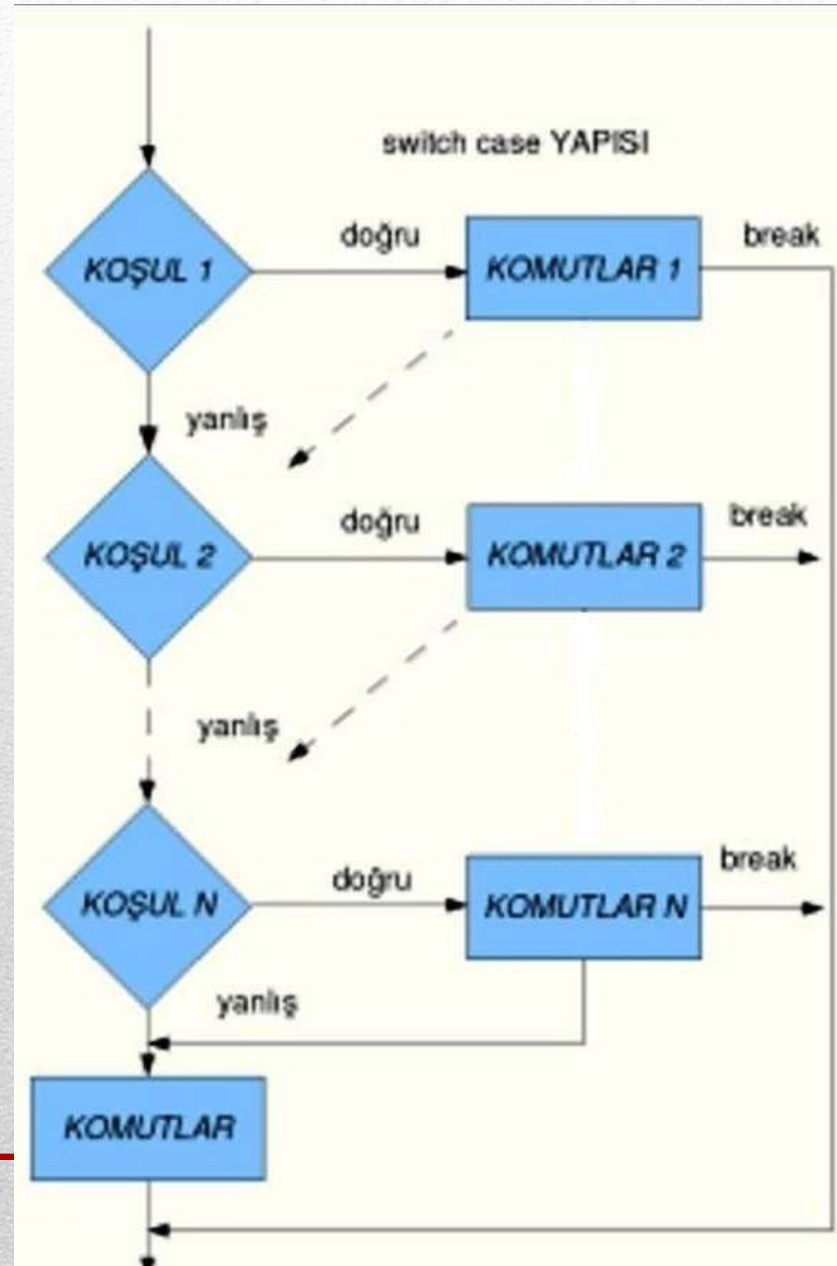
```
    case sabitn:  
        deyimn  
        break;
```

```
    default:  
        deyimd;
```

değişken değeri herhangi bir sabite eşit değilse **default** sabitinin deyimini **deyim**_d çalışır.

```
}
```

Switch – Case Akış Şeması



Örnek (Karşılaştırma)

- Girilen il plaka koduna göre ekrana ilgili şehir ismi yazdırılacaktır.
 - ANKARA
 - İSTANBUL
 - İZMİR
 - MANİSA
- Bunların dışında bir plaka kodu girilirse «TANIMSIZ PLAKA KODU» şeklinde hata mesajı verilecektir.

Örnek (Karşılaştırma)

```
int main()
{
    int plaka_kodu;
    printf("İlin plaka kodunu giriniz: ");
    scanf("%d", &plaka_kodu);

    switch (plaka_kodu) {
        case 6:
            printf("ANKARA\n");
            break;
        case 34:
            printf("İSTANBUL\n");
            break;
        case 35:
            printf("İZMİR\n");
            break;
        case 45:
            printf("MANİSA\n");
            break;
        default:
            printf("TANIMSIZ PLAKA KODU\n");
    }

    return 0;
}
```

```
int main()
{
    int plaka_kodu;
    printf("İlin plaka kodunu giriniz: ");
    scanf("%d", &plaka_kodu);

    if (plaka_kodu==6)
        printf("ANKARA\n");
    else if (plaka_kodu==34)
        printf("İSTANBUL\n");
    else if (plaka_kodu==35)
        printf("İZMİR\n");
    else if (plaka_kodu==45)
        printf("MANİSA\n");
    else
        printf("TANIMSIZ PLAKA KODU\n");

    return 0;
}
```


Switch-Case Kullanım Hataları

- **break** deyiminin kullanılmasının unutulması.
- switch-case bloğundaki **{ }** sembollerinin kullanımlarının unutulması.
- **default** deyiminin ve içeriğinin kullanımının unutulması.
- Kontrol edilen değişken ile verilen sabitlerin farklı tipte olması.
- **int** ve **char** veri türleri dışında değişken tanımlamaya çalışmak (double, string)

If-Else ve Switch-Case Karşılaştırması

1. İşlevsellik:

- C programlama dilinde
 - switch-case deyimi sadece **char** ve **int** veri türlerini destekler. Bu durum if-else deyimine göre dezavantajdır.
 - switch-case deyimlerinde aralık kontrolü ($x > 1000$) yapılamaması yine bir dezavantajdır.

2. Performans:

- Her iki karşılaştırma deyiminin kullanımını farklı derleyicilerde ve farklı programlama dillerinde farklı performans sonuçları üretebilir.
- Genelde **switch-case** kullanımını **daha performanslı** çalışır.

3. Okunabilirlik:

- **switch-case** deyimlerinin kod okunabilirliği her zaman daha yüksektir.

```
/**
 * Code Readability
 */
if (readable()) {
    be_happy();
} else {
    refactor();
}
```


ÖDEV

- Ödev 1:

Kullanıcıdan 2 sayı girmesini ve yaptırmak istediği matematiksel işlemin işaretini girmesini isteyiniz ve istenen işlem sonucunu ekrana yazdırınız (If-Else yapısıyla).

Örnek ekran çıktısı:

Birinci sayiyi giriniz: 5

İkinci sayiyi giriniz: 6

Yapilacak islemi seciniz (+, -, *, /): +

Islem Sonucunuz: 5+6=11

- Ödev 2:

Switch-case yapısıyla kullanıcının girdiği harflere göre 4 takımın ismini ekrana yazdırınız.

b,B=>Besiktas f,F =>Fenerbahce g,G =>Galatasaray k,K =>Konyaspor