

# İskenderun Teknik Üniversitesi



**Fakülte:** Mühendislik ve Doğa Bilimleri *Fakültesi*

**Bölüm:** *Bilgisayar Mühendisliği Bölümü*

**Ders:** Algoritmalar ve Programlama

**Dönem:** 2020 – 2021 (Güz)

**Öğretim üyesi:** Dr. Öğr. Üyesi Mehmet SARIGÜL

5.

***Hafta***

- Sıralı yapılar:
  - Varsayılan olarak sırayla yürütülen programlar
- Seçim yapıları:
  - if, if... else ve switch
- Tekrar yapıları:
  - while, do ... while and for

# Yapılar

---



- Kontrol yapıları
- Birleşik ifadeler
  - sırayla çalıştırılan, {ve} ile parantez içine alınmış bir ifade grubu

# Kontrol yapıları

---

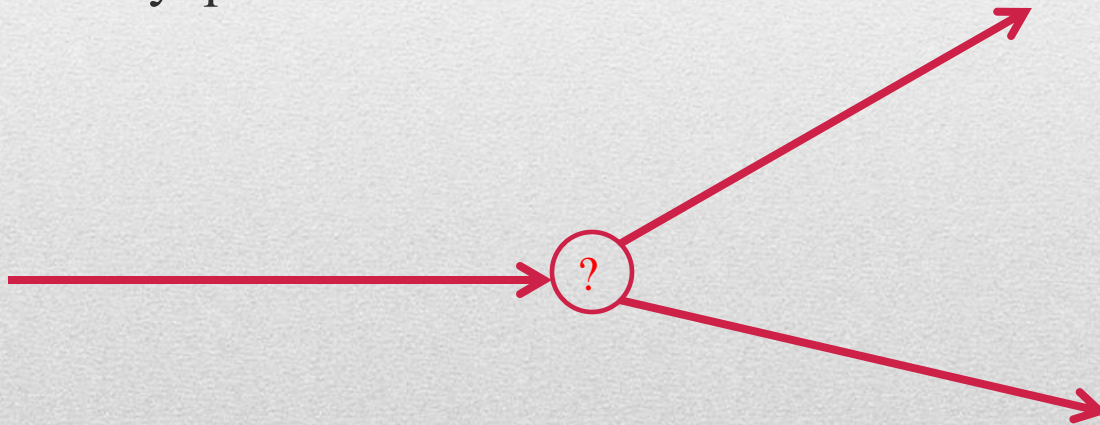
```
{  
    ifade;  
    ifade;  
    .  
    .  
    .  
    ifade;  
}
```

# Birleşik ifadeler

---



- seçim kontrol yapısı
  - alternatif program ifadeleri arasından seçim yapan bir kontrol yapısı



# Kontrol yapıları

---

- yanlış olan bir ifade
    - 0 ile temsil edilir
  - doğru
    - genellikle 1 ile temsil edilir
- `rest_heart_rate > 75`

# Koşullar

---



| Operator | Anlamı     | Tipi      |
|----------|------------|-----------|
| <        | küçüktür   | ilişkisel |
| >        | büyüktür   | ilişkisel |
| <=       | Küçük eşit | İlişkisel |
| >=       | Büyük eşit | İlişkisel |
| ==       | Eşit       | Eşitlik   |
| !=       | Eşit değil | Eşitlik   |

# İlişkisel ve Eşitlik Operatörleri

- İlişkisel operatörler  $<$ ,  $>$ ,  $\leq$  ve  $\geq$  'dir.
- İşlenenler olarak 2 ifade alır
- Ya int değeri 0 (yanlış) ya da int değeri 1 (doğru) verir.
  
- Örnek:  $a = 1$ ,  $b = 2$  varsayalım.
- İfade Değeri
  - $a \leq b$ , 1 verir
  - $a < b - 5$ , 0 verir
  - $a + 10 / b \leq -3 + 8$ , 0 verir

# İlişkisel operatörler



- $a == b$  ve  $a = b$  ifadelerinin görsel olarak benzer olduğuna dikkat edin!!!!
- İfadeler
  - $a == b$  eşitlik için bir testtir.
  - $a = b$  bir atama ifadesidir

# Eşitlik ifadeleri

- mantıksal ifadeler
- mantıksal operatörlerden birini veya birkaçını kullanan bir ifade
  - && (ve)
  - || (veya)
  - ! (değil)

# Mantıksal Operatorler

---




- && veya || ile bağlantılı ifadeler soldan sağa doğru değerlendirilir.
- mantıksal tümleme (olumsuzluk) :!
  - koşulun değeri 0 (yanlış) olduğunda koşulun tamamlayıcısı 1 (doğru) değerine sahiptir
  - koşulun değeri sıfır olmadığında (doğru) koşulun tamamlayıcısı 0 (yanlış) değerine sahiptir

# Mantıksal ifadeler

| ifade      | değer |
|------------|-------|
| $!5$       | 0     |
| $!!5$      | 1     |
| $!(6 < 7)$ | 0     |
| $!6 < 7$   | 1     |
| $!(3-4)$   | 0     |

# Mantıksal ifadeler



| Operator               | Öncelik   |
|------------------------|---|
| ()                     | En yüksek   |
| ! + - (unary operator) |  |
| * / %                  |   |
| + -                    |   |
| < <= >= >              |   |
| == !=                  |   |
| &&                     |   |
|                        |   |
| =                      | En düşük  |

# Operatör Önceliği

int i=5, j=10, k=20;

$i > 5 * k \% 3 \ \&\& \ !(\underline{k-2} != 18)$

↓

$i > 5 * k \% 3 \ \&\& \ !(\underline{18} != 18)$

↓

$i > 5 * k \% 3 \ \&\& \ !\underline{0}$

↓

$i > \underline{5 * k} \% 3 \ \&\& \ 1$

↓

$i > \underline{100 \% 3} \ \&\& \ 1$

↓

$\underline{i > 1} \ \&\& \ 1$

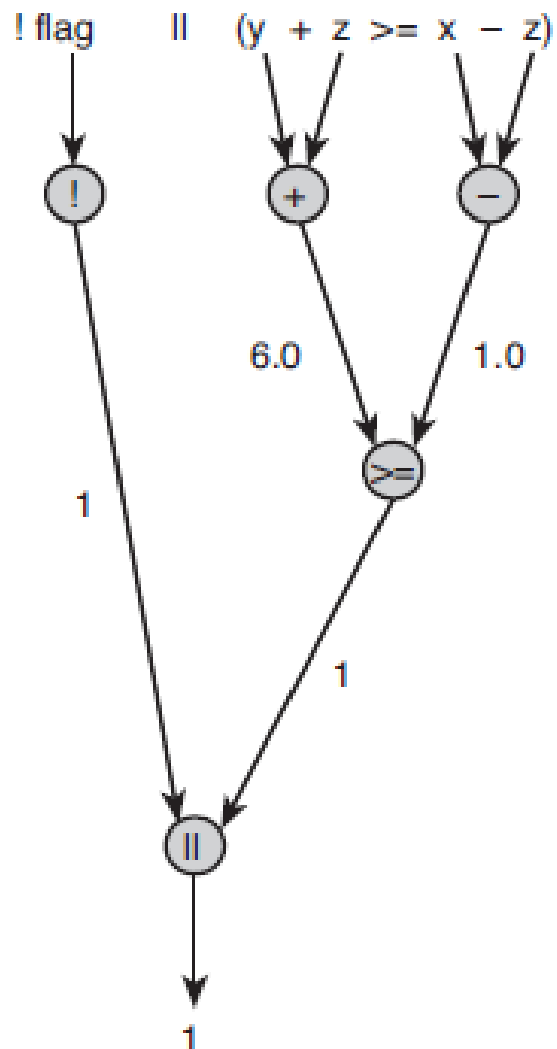
↓

$\underline{1 \ \&\& \ 1}$

↓

1





| flag | y   | z   | x   |
|------|-----|-----|-----|
| 0    | 4.0 | 2.0 | 3.0 |

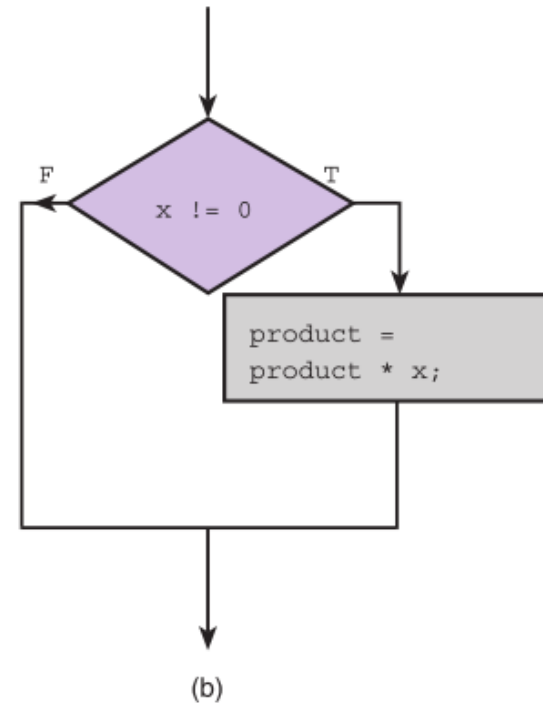
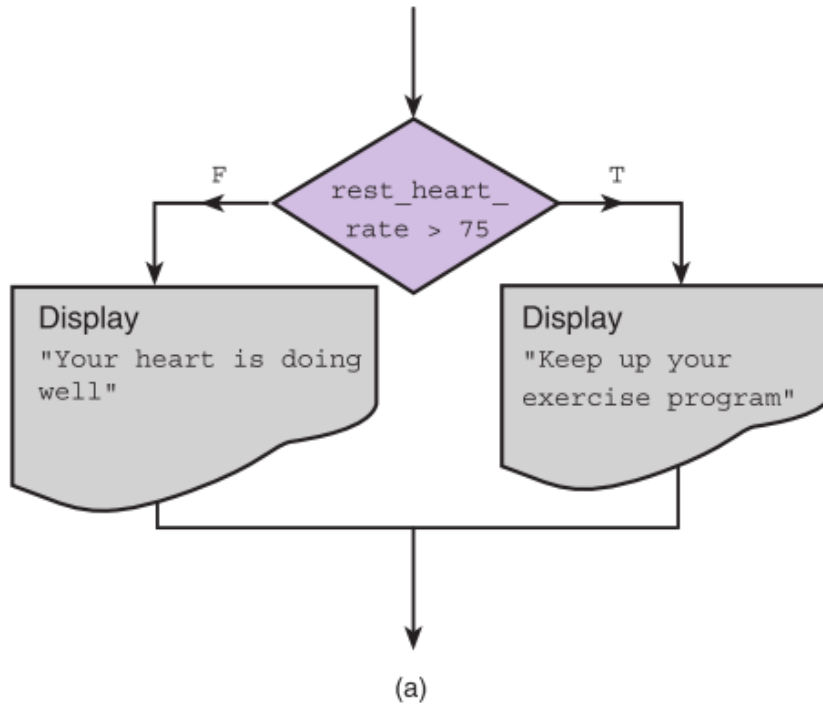
  

|          |  |                |    |                |
|----------|--|----------------|----|----------------|
| !flag    |  | (y + z         | >= | x - z)         |
| <u>0</u> |  | <u>4.0 2.0</u> |    | <u>3.0 2.0</u> |
| 1        |  | <u>6.0</u>     |    | <u>1.0</u>     |
|          |  |                |    | <u>1</u>       |
| 1        |  |                |    |                |

- $\&\&$  ve  $\parallel$  işlenenlerini içeren ifadeler için, doğru veya yanlış sonucu bilindiğinde ifade işlemi durur.
- İfade1'in 0 olduğunu varsayalım.
  - $\text{ifade1} \&\& \text{ifade2} = 0$  (ifade2 değerlendirilmeyecektir.)
- İfade1'in sıfır olmadığını varsayalım.
  - $\text{ifade1} \parallel \text{ifade2} = 1$  (ifade2 değerlendirilmeyecektir.)

# Kısa Devre Değerlendirmesi





# İki Alternatifli ve Bir Alternatifli if İfadelerinin Akış Şemaları

- FORM: if (koşul)  
ifade\_T;
- ÖRNEK: if (x > 0)  
prod = prod \* x;
- YORUM: Koşul doğru olarak değerlendirilirse (sıfır olmayan bir değer), o zaman ifade\_T yürütülür; aksi takdirde, ifade\_T atlanır.

# İf Seçim İfadesi



- FORM: if (koşul)  
ifade\_T;  
else  
ifade\_F;
- ÖRNEK: if (x > 0)  
printf ("pozitif \ n");  
else  
printf ("negatif \ n");
- YORUMLAMA: Koşul doğru olarak değerlendirilirse (sıfır olmayan bir değer), o zaman ifade\_T çalıştırılır ve ifade\_F atlanır; aksi takdirde, ifade\_T atlanır ve ifade\_F çalıştırılır.

# If else seçim ifadesi

- if
  - Yalnızca koşul doğruysa bir eylem gerçekleştirir
- if...else
  - Hem koşul doğru hem de yanlış olduğunda gerçekleştirilecek bir eylemi belirtir



- Birleşik ifade:
  - Bir çift kaşlı ayraç içindeki ifadeler kümesi
- Örnek:
- `if (derece >= 60)`

```
printf ("Geçti. \n");  
else  
{  
printf ("Başarısız. \n");  
printf ("Bu kursu tekrar almalısınız. \n");  
}
```

- Üçlü koşullu operatör (? :)
- Üç bağımsız değişken alır (koşul, doğruysa değer, yanlışsa değer)
- `derece >= 60 ? printf ("Geçti \n") : printf ("Başarısız \n");`



- if... else seçim ifadelerini if ... else seçim ifadesinin içine yerleştirerek birden fazla durumu test edin
- Koşul karşılandığında, ifadelerin geri kalanı atlanır
- Pratikte genellikle derin girinti kullanılmaz

# İç içe if/else

---

```
If not >= 90 ise
    Print "A"
else
    If not >= 80 ise
        Print "B"
    else
        not >= 70 ise
            Print "C"
        else
            If not >= 60 ise
                Print "D"
            else
                Print "F"
```



```
if (expr1)
    statement1
else if(expr2)
    statement2
else if(expr3)
    statement3
.....
else if(exprN)
    statementN
else
    default statement
next statement
```

# İç içe if/else

```
if(grade >= 90)
    cout<<"A";
else if (grade >= 80)
    cout<<"B";
else if (grade >= 70)
    cout<<"C";
else if (grade >= 60)
    cout<<"D";
else
    cout<<"F";
```



- birkaç alternatiften birini seçmek için kullanılır
- seçim değerine dayandığında kullanışlıdır
  - tek değişken
  - veya basit bir ifade
- değerler int veya char türünde olabilir
  - double,float değil

# Anahtar Çoklu Seçim Yapısı

---

```
switch ( a_variable ){  
  case value1:  
    actions;  
    break;  
  case value2 :  
    actions;  
    break;  
  ...  
  default:  
    actions;  
}
```

# Switch yapısı

---



- Çoklu seçim yapısını kullanarak basit bir hesap makinesi yazın

# Egzersiz

---





# Ders Sonu

