

# Akış Kontrolü

Ders 4

# Tekrar

---

- ▶ Değişkenler hafızada saklı değerler için kullanılan isimlendirilmiş referanslardır.
- ▶ Değişken tanımlama( deklarasyon )
  - ▶ Değişken tipi değişken adı
  - ▶ `int _a ;`
  - ▶ `char character;`
  - ▶ `long int bigNumber;`
- ▶ Değişken Başlatma
  - ▶ Tanımlanmış değerlere değer atama
    - `int _a; /* _a tanımlandı */`
    - `long int bigNumber; /* bigNumber tanımlandı */`
    - `_a = 5; /* _a başlatıldı */`
    - `bigNumber = 178872228; /* bigNumber başlatıldı */`

# Tekrar

---

- ▶ Aritmetik operatörler (işlemler)
- ▶ İlişkisel operatörler
- ▶ Mantiki (Logical) operatörler

# Tekrar

---

- ▶ Aritmetik operatörler: + - \* /
- ▶ Aritmetik ifadelerin tipleri
  - ▶ Eğer bir ifade sadece aynı tip değişken ve sabitlerden müteşekkil ise,
    - ▶ o ifadenin tipide değişken ve sabitlerin tiplerinin aynısıdır.
    - ▶  $5 / 5 = 1$ ,  $16 / 5 = 3$ ,  $32 / 33 = 0$
  - ▶ Eğer farklı tip değişken veya sabit içeriyorsa;
    - ▶ Küçük tipler büyük tiplere yükseltilir sonra işlem yapılır.
    - ▶ int tipi float olur.
    - ▶ float da double olur.

# Tekrar

---

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

```
int main () {
```

```
    int firstNumber = 9;
```

```
    int secondNumber = 5;
```

```
    float thirdNumber = 2.2;
```

```
    printf("Ifadenin sonucu %d \n",  
           firstNumber*secondNumber/thirdNumber);
```

```
    printf("Ifadenin sonucu %f \n",  
           firstNumber*secondNumber/thirdNumber);
```

```
    return (0);
```

```
}
```

# Tekrar

---

- Tip Çevirme: Parantez içerisine tip ismini yazarak bir tipi diğerine dönüştürebilirsiniz:

```
int firstNumber = 9;
int secondNumber = 5;
float thirdNumber = 2.2;
/* bu, 20 yazdırır */
printf(" Ifadenin sonucu: %d \n",
(int)(firstNumber*secondNumber/thirdNumber));

/* bu, 0 veya beklenmedik bir sonuc yazdırır */
printf(" Ifadenin sonucu: %d \n",
(firstNumber*secondNumber/thirdNumber));

/* bu 20.454544 yazdırır */
printf(" Ifadenin sonucu: %f \n",
(firstNumber*secondNumber/thirdNumber));
```

# Tekrar

---

- ▶ Ondalıklı sayıları printf de yuvarlama:
- ▶ `printf("%.2f \t", 3.145677);`
  - ▶ Prints 3.15
- ▶ `printf("%.2f \t", 3.1449);`
  - ▶ Prints 3.14
- ▶ `printf("\n%.3f \t", 3.145677);`
  - ▶ ?
- ▶ `printf("%.3f \t", 3.144989);`
  - ▶ ?

# Tekrar

---

- ▶ Boolean İfadeler: Doğru veya yanlış ( true-false) şeklinde bir sonucu olan ifadeler: (C de, 0 veya 1).
- ▶ İlişkisel Operatörler
  - ▶ >
    - ▶ Büyük
    - ▶  $a > b$ 
      - Eğer a büyük b ise,  $(a > b)$  ifadesi 1 değeriyle sonuçlanır.
      - Eğer a büyük değilse, ifade 0 değeriyle sonuçlanır.
  - ▶ >=
    - ▶ Büyük eşit
  - ▶ <
    - ▶ Küçük
  - ▶ <=
    - ▶ Küçük eşit
  - ▶ ==
    - ▶ Eşit
  - ▶ !=
    - ▶ Eşit değil (Farklı)



# Tekrar

---

```
int a1 = 3;  
int a2 = 5;  
printf("%d\t%d\t%d", a1 == a2, a1 > a2, a1 < a2);
```

# Mantiki İşlemler (Logical Operations)

---

## ▶ &&

▶ Ve işlemi

▶  $(a \&\& b) = ?$

a	b	$(a \&\& b)$
0	0	0
0	1	0
1	0	0
1	1	1

## ▶ ||

▶ Veya işlemi

▶  $(a || b) = ?$

a	b	$(a    b)$
0	0	0
0	1	1
1	0	1
1	1	1

# Mantiki İşlemler (Logical Operations)

---

▶ !

- ▶ Değil işlemi
- ▶ !(0) ifadesi 1 dir.
- ▶ !(1) ifadesi 0dır.

```
int x = 5, y = 4;  
printf("%d %d\n",  
        !((x == y) && (x != y)), ((x != 5) || (x > y + 4)));
```

# Bloklar

---

- ▶ `{ }` kullanılarak çoklu cümlecikler, blok denilen tek bir ünite içine konulabilirler.
- ▶ Bloklar iç içe gömülebilirler.
- ▶ İçteki gömülü blok dıştaki bloktan tek bir cümlecik olarak görülür.
- ▶ Gömülü blok kendi değişkenleri için lokal bir kapsam oluşturur.
  - ▶ İçteki blokta tanımlı değişkenler bu bloğa lokaldirler.
  - ▶ Dıştaki bloktan erişilemez kullanılamazlar.

# Bloklar

---

```
{  
    int b1 = 0;  
    /* sadece b1 görünmekte */  
    {  
        int b2 = 0;  
        /* b1 ve b2 görünmekte */  
        {  
            int b3 = 0;  
            /* b1, b2, ve b3 görünmekte */  
        }  
        /* b1 ve b2 görünmekte */  
    }  
    /* sadece b1 görünmekte */  
}
```

# Bloklar: Girintiler kodumuzu daha okunabilir yapar.

---

```
{
  int b1 = 0;
  /* sadece b1 görünmekte */
  {
    int b2 = 0;
    /* b1 ve b2 görünmekte */
    {
      int b3 = 0;
      /* b1, b2, ve b3 görünmekte */
    }
    /* b1 ve b2 görünmekte */
  }
  /* sadece b1 görünmekte */
}
```

# Değişken gölgeleme (Overshadowing)

---

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

```
int main()
```

```
{
```

```
    int x = 1; /* bloğun dışında */
```

```
    /* birleşik cümlecikler */
```

```
{
```

```
    int x = 1; /* bloğun içinde */
```

```
    x = x + 1;
```

```
    printf("1.printf: %d\n", x); /* prints x */
```

```
}
```

```
printf("2.printf: %d\n", x); /* prints x */
```

```
return 0;
```

---

```
}  
5
```

# Çeşitli Blok Yapıları

---

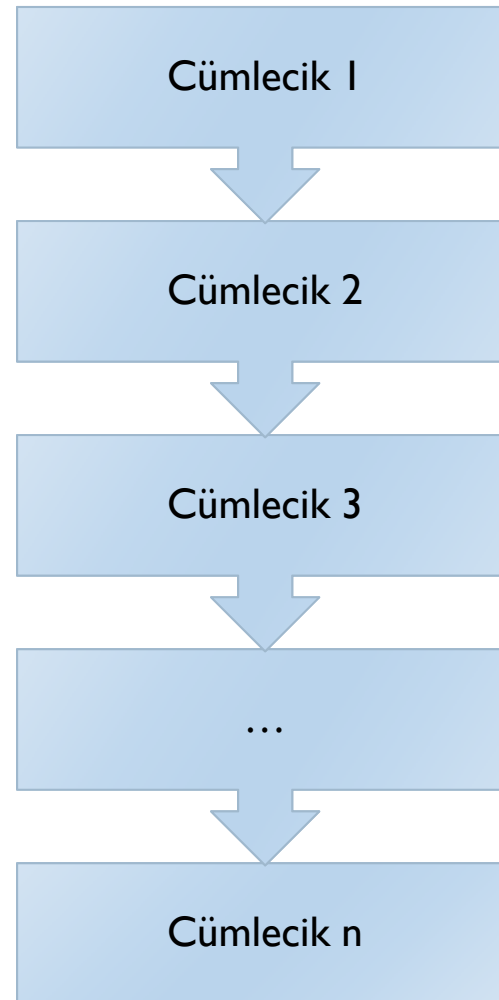
- ▶ **Şartlı Bloklar**
  - ▶ Çalışması bir şarta bağlı bloklar
- ▶ **Tekrarlı Bloklar**
  - ▶ Çalışması belirli sayıda tekrarlanan bloklar



# Akış Kontrol

---

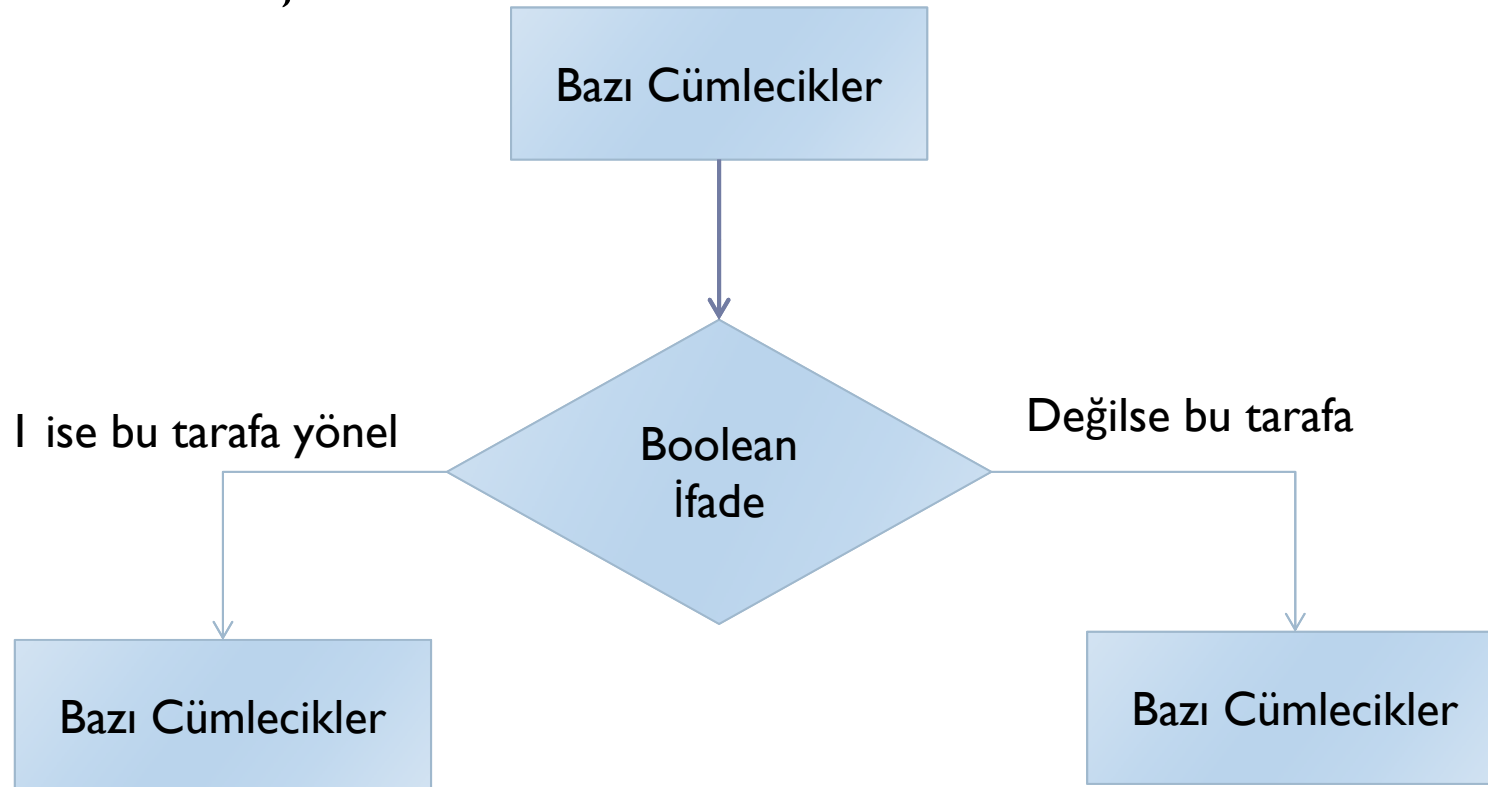
## ► Sıralı Çalıştırma



# Akış Kontrolü

---

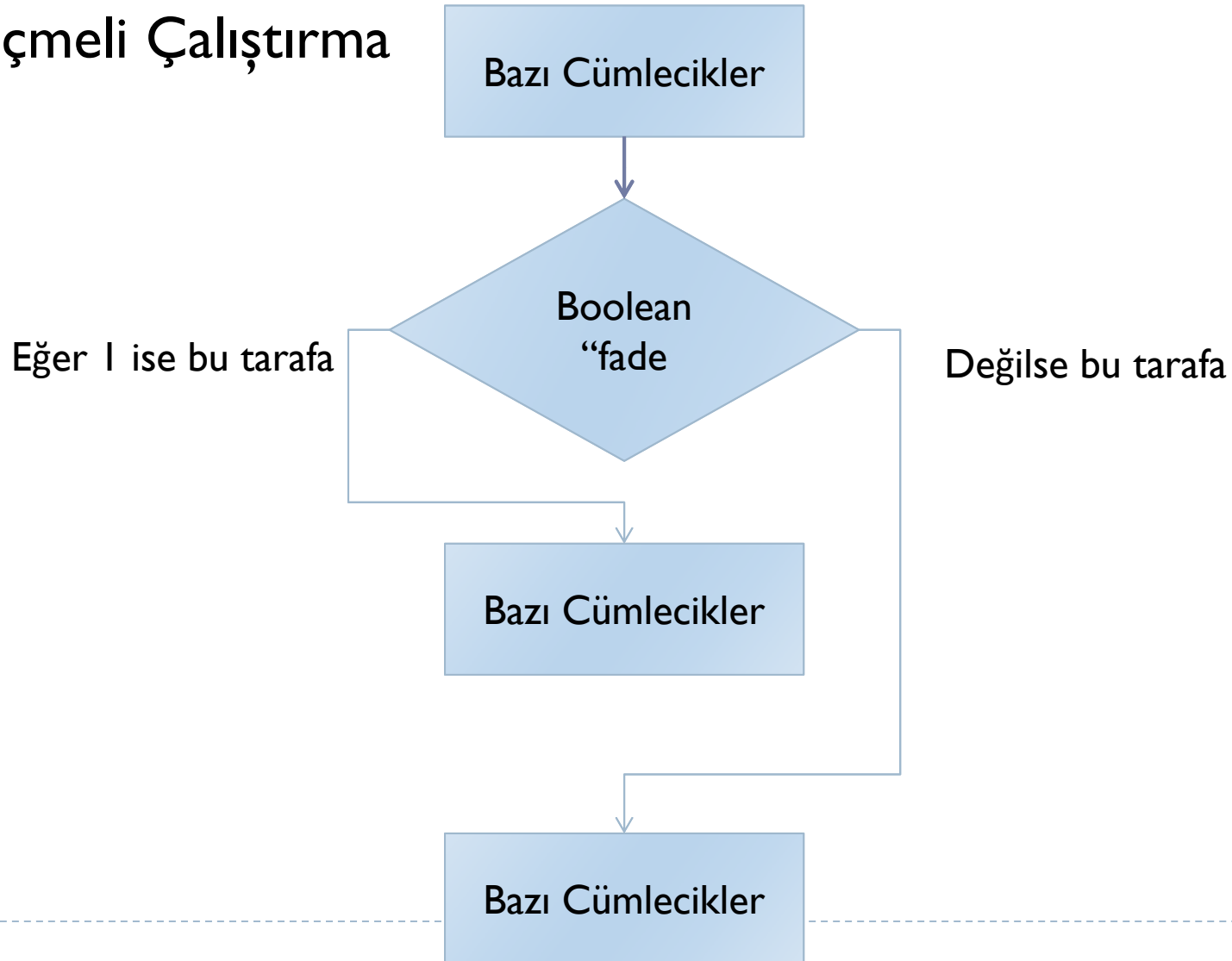
## ► Seçmeli Çalıştırma



# Akış Kontrol

---

## ► Seçmeli Çalıştırma



# if Cümlecikleri

---

- ▶ Eğer bir ifade 1 ise (yani doğru-true), o zaman takip eden cümlecigi çalıştır.

- ▶ if (Boolean İfade)

Cümlecik1;

**if** (a > b)

**printf** ("a buyuktur b\n");

- ▶ Eğer bir ifade 1 ise, takip eden bloğun içindeki cümlecikleri çalıştır (Şartlı blok - Conditional Block).

```
if (Boolean ifade){  
    Bazı Cümlecikler  
}
```

# if Cümlecikleri

---

```
#include <stdio.h>
int main() {
    char myc = 'A'; /* benim karakterim */
    char userc; /* kullanıcı tahmini */

    printf("\n Karakterimin ne olduğunu tahmin et: ");
    scanf("%c", &userc);

    if (myc == userc){
        printf (" Dogru bildiniz!\n");
    }
    return 0;
}
```

# Bunlar aynı mı?

---

```
if (value > 0 ){  
    result = 1.0;  
    printf("the result is %f",  
          result);  
}
```

```
if (value > 0 )  
    result = 1.0;  
    printf ("the result is %f",  
            result);
```

# if-else Cümlecikleri

---

- Eğer (if) bir şey true (doğru) ise, bazı cümlecikleri çalıştır; değilse (else) başka cümlecikleri çalıştır.

if (Boolean ifade)

{

    Bazı cümlecikler;

}

else

{

    Bazı cümlecikler;

}

# if-else Cümlecikleri

---

- Eğer (if) bir şey true (doğru) ise, bazı cümlecikleri çalıştır; değilse (else) başka cümlecikleri çalıştır.

```
#include <stdio.h>
int main() {
    char myc = 'A'; /* benim karakterim */
    char userc; /* kullanıcı tahmini */
    printf("\n Karakter tahmin et : ");
    scanf("%c", &userc);

    if (myc == userc){
        printf("Tahminin dogru!\n");
    }
    else
        printf("Tahminin yanlis!\n");

    return 0;
}
```



## Örnek

---

- ▶ Kullanıcıdan üç sayı girmesini isteyen ve bu sayıların en büyüğünü bulan bir program yazınız.

# Örnek

---

- ▶ Kullanıcıdan üç sayı girmesini isteyen ve bu sayıların en büyüğünü bulan bir program yazınız.
  - ▶ Giriş değerleri neler?
  - ▶ Çıkış değerleri neler?
  - ▶ Maksimumu hesaplamak için algoritmanın adımları neler?

# Örnek

---

- ▶ Kullanıcıdan üç sayı girmesini isteyen ve bu sayıların en büyüğünü bulan bir program yazınız.
  - ▶ Giriş değerleri neler?
    - ▶ Üç tane tam sayı
    - ▶  $n_1, n_2, n_3$  olsunlar
  - ▶ Çıkış değerleri neler?
    - ▶ Maksimum değerini tutmak için bir tam sayı değişken
  - ▶ Maksimumu hesaplamak için algoritmanın adımları neler?
    - ▶ Kullanıcıdan giriş değerlerinin alınması
    - ▶ İlk iki sayının karşılaştırılması ( $n_1$  ve  $n_2$ ).
    - ▶ Sonra büyük olanın üçüncü sayı ile karşılaştırılması ( $n_3$ ).
    - ▶ Maksimum değerın çıkış olarak verilmesi.

# Örnek

---

- ▶ Ders notlarındaki maksimum.c ye bakınız.

# if- else if .. else Cümlecikleri

---

```
if (Boolean ifade){  
    Bazı cümlecikler  
}  
else if (Boolean ifade){  
    Bazı cümlecikler  
}  
else if (Boolean ifade){  
    Bazı cümlecikler  
}  
...  
else {  
    Bazı cümlecikler  
}
```

# Örnek

- Kullanıcıdan almış olduğu notun harf karşılığını gösteren bir program yazınız.

Harf notu	Puanı
AA	90-100
BA	85-89
BB	80-84
CB	70-79
CC	60-69
DC	55-59
DD	50-54
FF	0-49

# Örnek

---

Harf notu	Puanı
AA	90-100
BA	85-89
BB	80-84
CB	70-79
CC	60-69
DC	55-59
DD	50-54
FF	0-49

- ▶ Giriş:
  - ▶ Kullanıcının puan notu
- ▶ Çıkış:
  - ▶ "AA", ... , "FF" den herhangi birisi
- ▶ Algoritma:
  - ▶ Karşılaştırmaların yapılması...
  - ▶ Ders notlarında harfnotu.c ye bakınız.

# İç içe (gömülü) if cümlecikleri

---

if (Boolean ifade)

{

    Bazı cümlecikler

    if(Boolean ifade)

    {

        Bazı cümlecikler

    }

    ..

    else

    {

        Bazı cümlecikler

    }

}

else

{

    Bazı cümlecikler

}



# switch-case Cümlecikleri

---

- ▶ Alternatif şartlı cümleler
- ▶ Giriş ifadesi olarak integer veya character
- ▶ İfadenin alabileceği farklı her bir değeri göz önüne alır:

```
switch(ifade){  
    case deger1:  
        Bazı cümlecikler  
        break;  
    case deger2:  
        Bazı cümlecikler  
        break;  
    ...  
    default:  
        bazı cümlecikler  
}
```

# switch-case Cümlecikleri

---

```
switch(ch){  
    case 'Y': /* if (ch == 'Y') */  
        printf("Yes!\n");  
        break;  
    case 'N': /* if (ch == 'N') */  
        printf("No!\n");  
        break;  
    default: /* otherwise */  
        printf("ch ne 'Y' ne de 'N' dir.\n");  
}
```

- ▶ Eşleşme bulunduğunda, takip eden break; cümleciğine kadar olan kodları çalıştırır.
- ▶ Eğer break; yoksa, kodun çalıştırılması sonuna kadar devam eder.

# Vaka Çalışması

---

- ▶ Basit bir hesap makinesi programı.
  - ▶ Kullanıcı işlemi ( +, veya -, veya \*, veya /) ve iki sayıyı girer.
  - ▶ Program işlemin sonucunu gösterir.
  - ▶ Ders notlarında hesapmakinesi.c

# Haftaya

---

- ▶ Döngü ve tekrarlar