



BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

DERS 9
Programlama Dilleri

Tarihsel bakış

- ▶ Eğer insanlar makine dilinde program yazmak durumunda kalsalardı bugün kullanılan birçok kompleks yazılımı programlamak imkansız hale gelecekti.
- ▶ Bu nedenle algoritmaların ifade edilebilmesini sağlayan birçok programlama dili geliştirilmiştir.

- ▶ Makine dili gibi sayısal olarak kodlanmış dillerde hata ayıklama işi oldukça zor ve yorucudur.
- ▶ 1940'lı yıllarda sayısal form yerine akılda daha kalıcı olan sembolik bir işaretleme sistemi geliştirdiler.

4056

MOV R5, R6

156C

166D

5056

306E

C000

LD R5,Price

LD R6,ShippingCharge

ADDI R0,R5 R6

ST R0,TotalCost

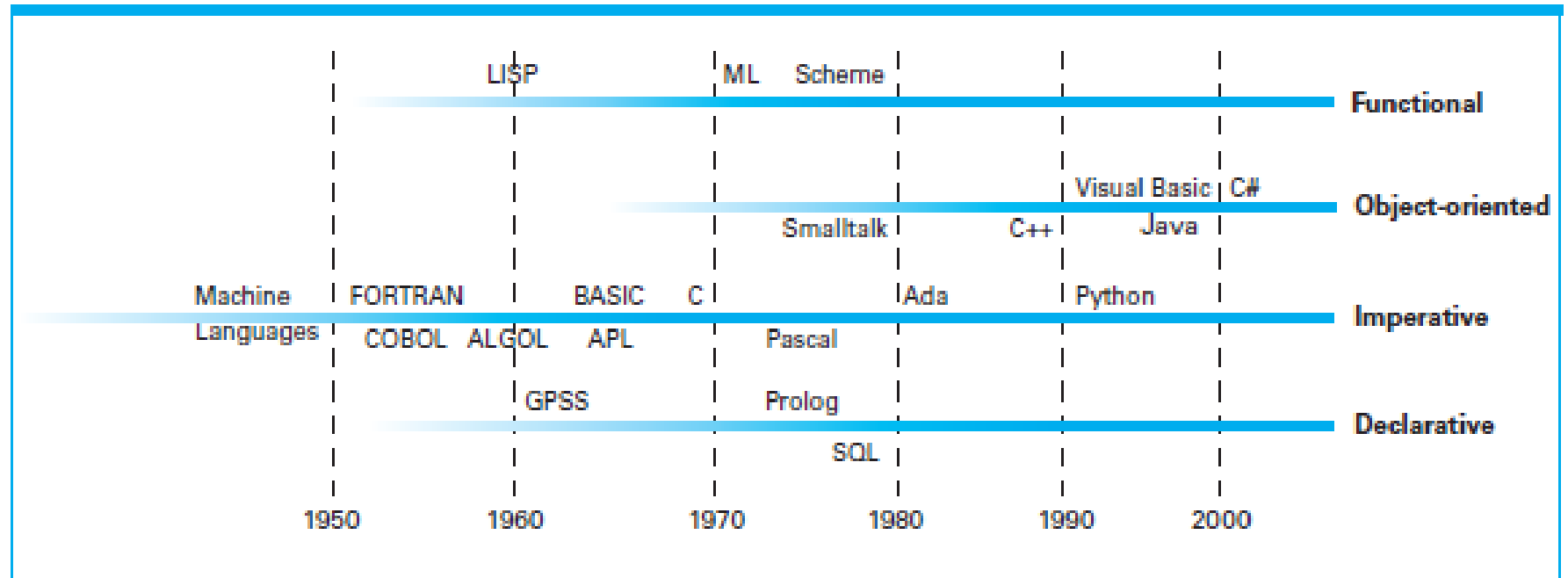
HLT

- ▶ Sembolik formdaki bu programlar **assembler** (çevirici) kullanılarak makine diline çevrilir.
- ▶ Programları temsil etmek için kullanılan sembolik sistemin tümü **assembly** dili olarak adlandırılır.
- ▶ Sonuçta **makine dili (birinci nesil diller)** ile **assembly dilinde (ikinci nesil)** temel öğeler aynıdır. Yani **assembly dili makine diline bağımlıdır.**

- ▶ Bu yüzden **3. nesil diller** (fortran, cobol) **makine bağımsız** olarak tasarlandı.
- ▶ 3. nesil dillerde yazılan kodlar **makine dilinde bir diziye karşılık** gelmektedir.
- ▶ Bu kodlar **compiler**(derleyici) ile makine diline dönüştürürler.

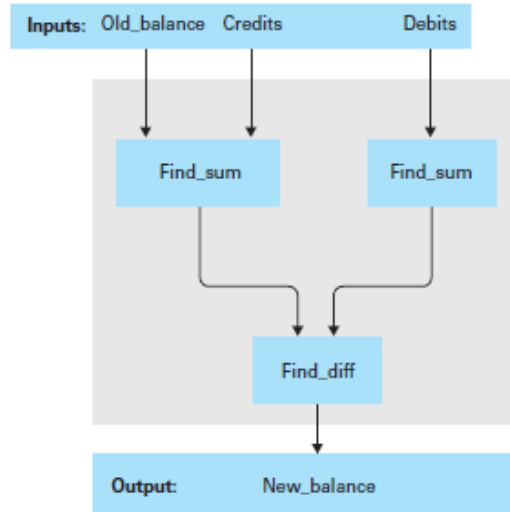
Programlama paradigmları

Figure 6.2 The evolution of programming paradigms



Fonksiyonel paradigma

`(Find_diff (Find_sum Old_balance Credits) (Find_sum Debits))`



`Total_credits = sum of all Credits`
`Temp_balance = Old_balance + Total_credits`
`Total_debits = sum of all Debits`
`Balance = Temp_balance - Total_debits`

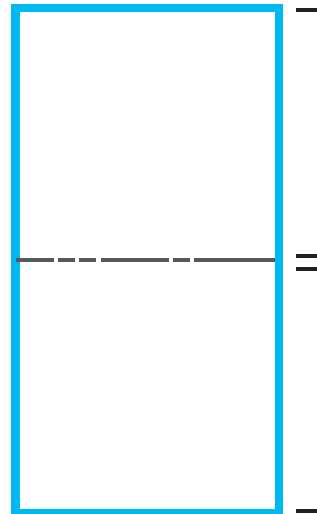
Geleneksel programlama kavramları

- ▶ Değişkenler ve veri türleri
- ▶ Veri türü: integer, float ..

```
int WeightLimit;  
int Height, Width;  
int WeightLimit = 100;
```

```
char Letter, Digit;  
float Length, Width;  
int Price, Tax, Total;  
char Symbol;
```

Program



The first part consists of declaration statements describing the data that is manipulated by the program.

The second part consists of imperative statements describing the action to be performed.

Veri yapısı

- Yaygın olarak kullanılan veri yapısı dizilerdir.

```
int Scores[2][9];
```

```
INTEGER Scores(2,9)
```

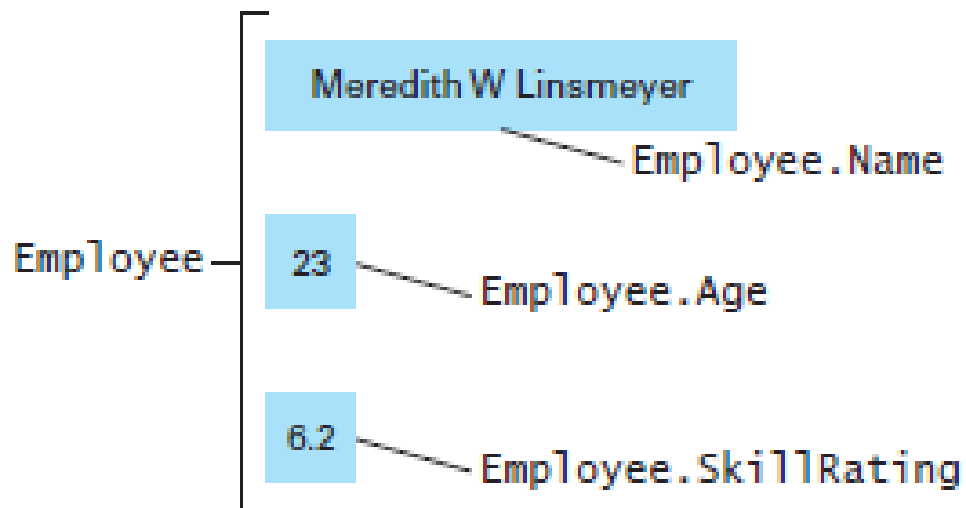
Scores

Scores (2,4) in
FORTRAN where
indices start at one.

Scores [1][3] in C
and its derivatives
where indices start
at zero.

struct

```
struct { char Name[25];  
        int Age;  
        float SkillRating;  
    } Employee;
```



Sabitler ve değişmezler

```
EffectiveAlt = Altimeter + 645
```

```
const int AirportAlt = 645;
```

```
final int AirportAlt = 645;
```

```
EffectiveAlt = Altimeter + AirportAlt
```

Atama işlemleri

$Z = X + Y;$

$Z := X + Y;$

$Z \leftarrow X + Y$

$'abra' + 'cadabra'$

Kontrol deyimleri

```
    goto 40
20  Evade()
    goto 70
40  if (KryptoniteLevel < LethalDose) then goto 60
    goto 20
60  RescueDamsel()
70  ...
```

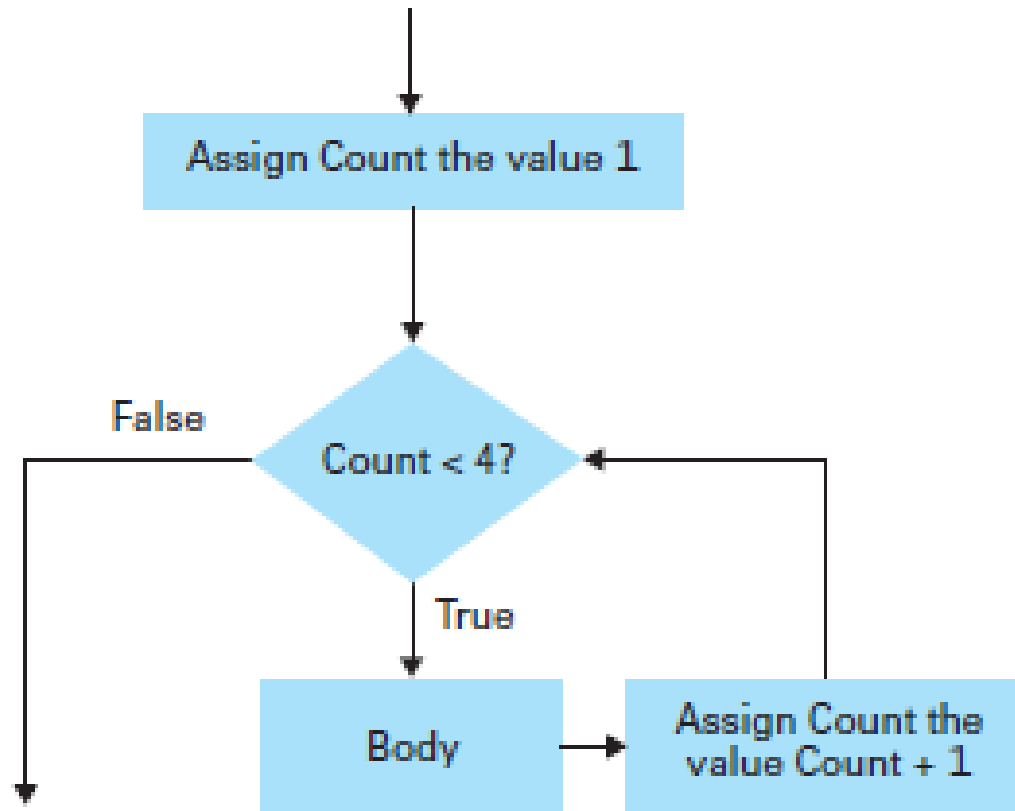
```
if (KryptoniteLevel < LethalDose):
    RescueDamsel()
else:
    Evade()
```

```
if (condition):  
    statementA  
else:  
    statementB
```

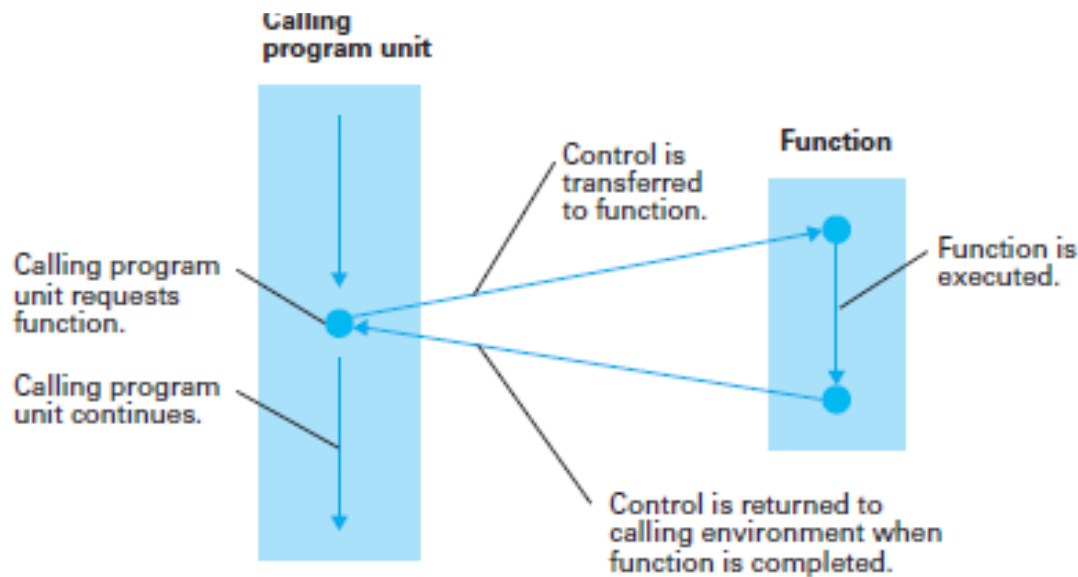
```
while (condition):  
    body
```

```
switch (variable) {  
    case 'A': statementA; break;  
    case 'B': statementB; break;  
    case 'C': statementC; break;  
    default: statementD;}
```

```
CASE variable IS  
    WHEN 'A'=> statementA;  
    WHEN 'B'=> statementB;  
    WHEN 'C'=> statementC;  
    WHEN OTHERS=> statementD;  
END CASE;
```



```
for (int Count = 1; Count < 4; Count++)  
    body ;
```



```
void ProjectPopulation (float GrowthRate)
```

```
{ int Year; } This declares a local variable named Year.
```

```
Population[0] = 100.0;
for (Year = 0; Year <= 10; Year++)
Population[Year+1] = Population[Year] + (Population[Year] * GrowthRate);
}
```



```
def PrintCheck(Payee, Amount):  
    PrintCheck('John Doe', 150)  
    PrintCheck(150, 'John Doe')
```

Verimli fonksiyonlar

```
def CylinderVolume(Radius, Height):  
    Volume = math.pi * Radius * Radius * Height  
    return Volume
```

```
float CylinderVolume (float Radius, float Height)
```

```
{ float Volume;
```

Declare a
local variable
named Volume.

```
Volume = 3.14 * Radius * Radius * Height;
```

```
return Volume;
```

Compute the volume of
the cylinder.

```
}
```

Terminate the function and
return the value of the
variable Volume.

```
Cost = CostPerVolUnit * CylinderVolume(3.45, 12.7);
```

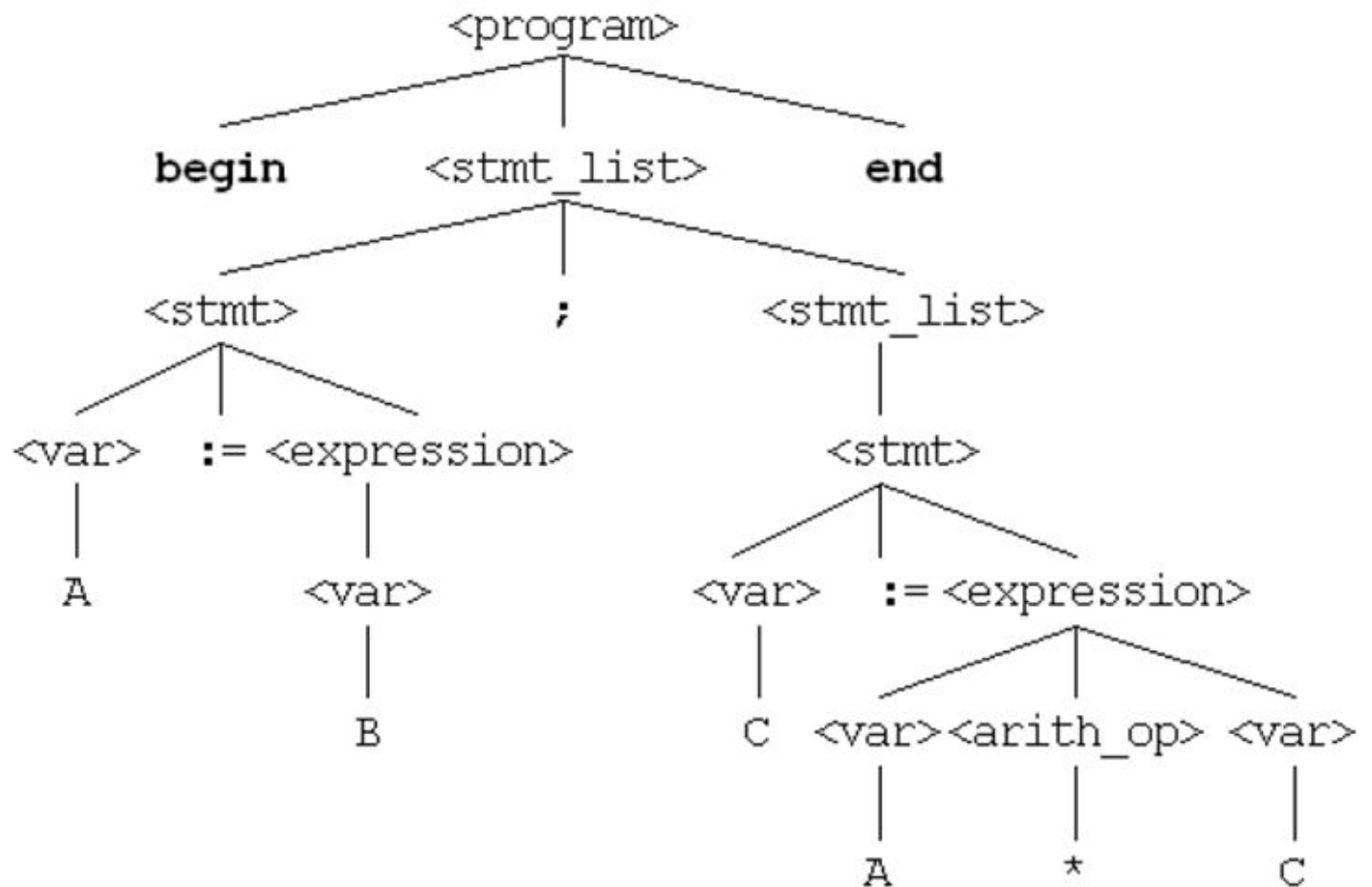
Algoritma Kavramı

- ▶ Algoritma; bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, **açıkça belirlenmiş** bir son durumunda sonlanan, sonlu işlemler kümesidir.
- ▶ Dur komutu alınmadığı sürece aşağıdaki adımları çalıştır:
 - ▶ Bir komut getir
 - ▶ Komutu çöz
 - ▶ Komutu yürüt

Ayrıştırma Ağacı



```
begin
  A:=B
  C:=A*B
end
```





Ders bitti

Erciyes Üniversitesi
Selçuk Üniversitesi
Sakarya Üniversitesi
Kahramanmaraş Sütçü İmam Üniversitesi
ders notları kaynak ve içerik olarak kullanılmıştır.