

# Nesneye Yönelimli Programlama – BLM 205

## Hafta 6: Kalıtım



Fenerbahçe Üniversitesi

# Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent  
Ofis: 311  
Email: [emre.levent@fbu.edu.tr](mailto:emre.levent@fbu.edu.tr)

Asistan: Arş. Gör. Uğur Özbalkan  
Ofis: 307  
Email: [ugur.ozbalkan@fbu.edu.tr](mailto:ugur.ozbalkan@fbu.edu.tr)

Asistan: Arş. Gör. Ecenur Alioğulları  
Ofis: 307  
Email: [ecenur.aliogullari@fbu.edu.tr](mailto:ecenur.aliogullari@fbu.edu.tr)

# Ders Planı

- Kalıtım
  - Kalıtım Temelleri
  - Kalıtım Türleri

# Kalıtım

- Kalıtım, daha önceden yaratılmış bir sınıfın tüm değişken ve fonksiyonlarını başka bir sınıfta tekrar kullanımını sağlayan bir mekanizmadır.
- Mevcut sınıfa'ya base class, yeni sınıfa ise, derived (türetilmiş) class denir.
- Türetilmiş sınıflar, türetildikleri sınıflara göre daha özelleşmiş objeleri temsil ederler.
- Örneğin araçlar sınıfı ve otomobiller sınıfı

# Kalıtım

- Bir sınıf, başka bir sınıftan doğrudan türetilebileceği gibi, dolaylı da türetilmiş olabilir.

# Kalıtım

## Örnek 1:

- Araç sınıfından türetilmiş bir otomobil sınıfı olsun, otomobil sınıfından da Volkswagen sınıfı olsun. Volkswagen sınıfı araç sınıfından dolaylı üretilmiş olur.

# Kalıtım

## Örnek 2:

- Araç, kamyon ve otobüs isminde 3 farklı sınıf yaratılacaktır. Bu sınıfların fonksiyonlarından
  - yakitMiktariniAl()
  - yolcuKapasitesiniAl()
  - frenYap()

işlevleri aynı olabilir.

# Kalıtım

## Örnek 2:

- Kalıtım metodolojisini izlemeden, bu sınıfları ayrı ayrı yazıldığında aşağıdaki gibi tüm fonksiyonların kopyaları tüm sınıflarda olacaktır. Bir fonksiyonda değişiklik yapılma ihtiyacı olduğunda, diğerlerinde de değiştirmek gerekecektir.

### Araba

```
yakitMiktariniAl()  
yolcuKapasitesiniAl()  
frenYap()
```

### Kamyon

```
yakitMiktariniAl()  
yolcuKapasitesiniAl()  
frenYap()
```

### Otobus

```
yakitMiktariniAl()  
yolcuKapasitesiniAl()  
frenYap()
```



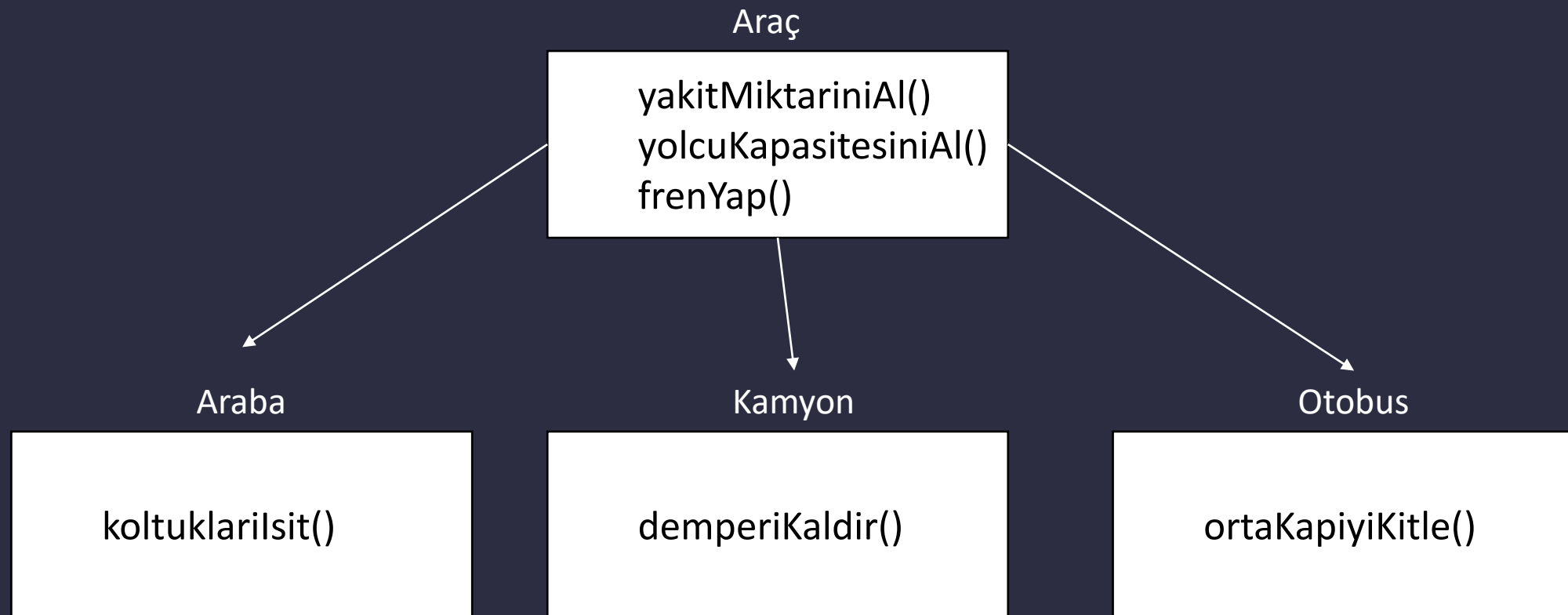
# Kalıtım

## Örnek 2:

- Bu durumdan kaçınılmak için kalıtım (inheritance) kullanılmalıdır.
- Tüm sınıfları kapsayacak araç isminde bir sınıf yapıp, bu ortak kullanılacak fonksiyonları bu sınıfa yazıp, diğer sınıfların (araba, otobüs ve kamyon) bu sınıftan kalıtım yapmalarını sağlatılmalıdır.

# Kalıtım

## Örnek 2:



# Kalıtım

## Python'daki Syntax'i

```
class sinifAdi (anaSinifAdi):
```

```
    #Degisken ve Fonksiyonlar
```

# Kalıtım

## Python'daki Syntax'ı

### Örnek Kod Parçacığı

```
class kisiler(object):  
  
    def __init__(self, isimArg):  
        self.isim = isimArg  
  
    def isimAl(self):  
        return self.isim  
  
class calisanlar(kisiler):  
  
    calismaSaati = 40  
    def calismaSaatiAl(self):  
        return calismaSaati  
  
kisi1 = kisiler("Ali")  
print(kisi1.isimAl())  
  
calisan1 = calisanlar("Veli")  
print(calisan1.isimAl(),  
      calisan1.calismaSaati)
```

### Çıktı

Ali  
Veli 40

Kisiler sınıfından kalıtım yapıldı

Constructor çağrıldığında, kisilerin  
constructor'u da çalıştırıldı

# Kalıtım

## Python'daki Syntax'ı

### Örnek Kod Parçacığı

```
class kisiler( object ):  
  
    def __init__(self, isim, kimlikNo):  
        self.isim = isim  
        self.kimlikNo = kimlikNo  
  
class calisanlar( kisiler ):  
    def __init__(self, isim, kimlikNo,  
gelir):  
        self.gelir = gelir  
        kisiler.__init__(self, isim,  
kimlikNo)  
  
    def goster(self):  
        print(self.isim)  
        print(self.kimlikNo)  
        print(self.gelir)  
  
a = calisanlar('Ali', 123456, 500)  
  
a.goster()
```

### Çıktı

Ali  
123456  
500

# Kalıtım

## Python'daki Syntax'i

Bir sınıf birden çok sınıftan kalıtım yapmış olabilir.

```
class sinifAdi (anaSinifAdi, anaSinifAdi2):
```

```
#Degisken ve Fonksiyonlar
```

# Kalıtım

## Python'daki Syntax'ı

### Örnek Kod Parçasığı

```
class anaSinif1(object):  
    def __init__(self):  
        self.metin1 = "Test"  
        print("Ana Sinif1")  
  
class anaSinif2(object):  
    def __init__(self):  
        self.metin2 = "Abc"  
        print("Ana Sinif2")  
  
class turetilmisSinif(anaSinif1, anaSinif2):  
    def __init__(self):  
  
        anaSinif1.__init__(self)  
        anaSinif2.__init__(self)  
        print("Turetilmis Sinif")  
  
    def goster(self):  
        print(self.metin1, self.metin2)  
  
obj = turetilmisSinif()  
obj.goster()
```

### Çıktı

```
Ana Sinif1  
Ana Sinif2  
Turetilmis Sinif  
Test Abc
```

# Kalıtım

## Python'daki Syntax'i

### Örnek Kod Parçasığı

```
class Base(object):  
  
    def __init__(self, isim):  
        self.isim = isim  
  
    def isimAl(self):  
        return self.isim  
  
class kalitimSeviye1(Base):  
  
    def __init__(self, isim, yas):  
        Base.__init__(self, isim)  
        self.yas = yas  
  
    def yasAl(self):  
        return self.yas
```

### Örnek Kod Parçasığı

```
class kalitimSeviye2(kalitimSeviye1):  
  
    def __init__(self, isim, yas, adres):  
        kalitimSeviye1.__init__(self, isim,  
yas)  
        self.adres = adres  
  
    def adresAl(self):  
        return self.adres  
  
obj = kalitimSeviye2("Test", 23, "Adres  
ABC")  
print(obj.isimAl(), obj.yasAl(),  
obj.adresAl())
```

Çıktı

Test 23 Adres ABC



# Objeler ve Sınıflar

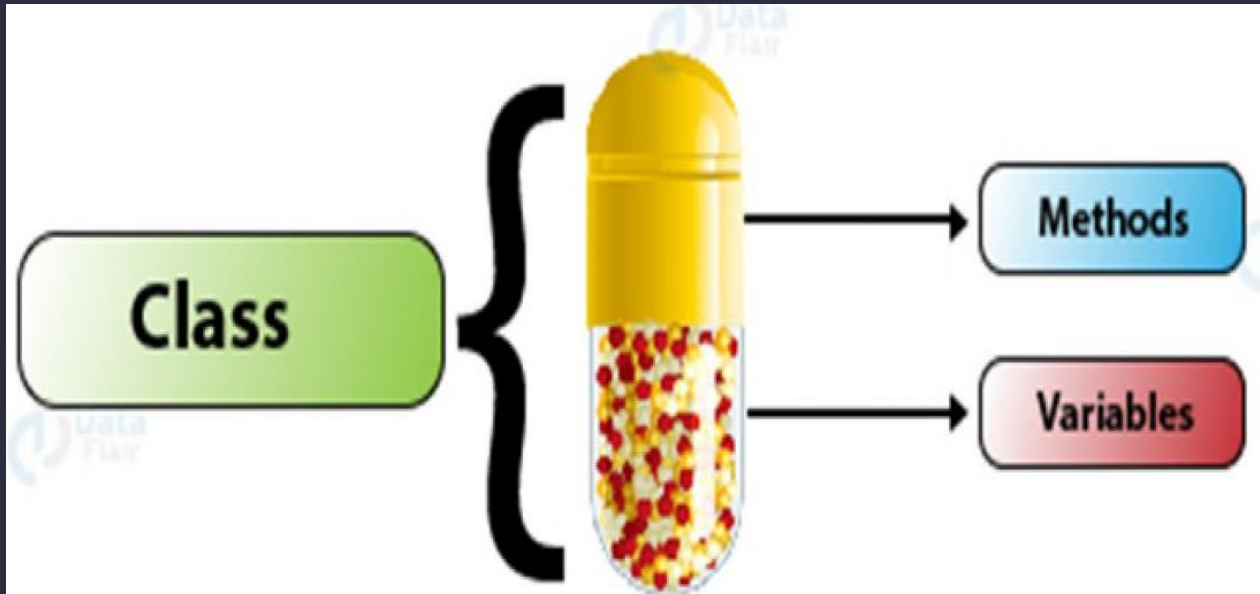
Private fonksiyon ve değişkenler:

Kalıtım yapan sınıfın, ana sınıftaki değişken veya fonksiyonlarından erişememesi için iki alt tire `__` işareti değişkenin veya fonksiyonun başına konur.

Dolayısıyla türetilmiş sınıf o değişken veya fonksiyona erişemez.

Bu yaklaşım genellikle, alt sınıfın, üst sınıftaki bir elemana doğrudan erişmesinin istenmediği durumlarda kullanılır. Bu yaklaşıma encapsulation denir.

# Objeler ve Sınıflar



Genellikle metodlar üzerinden değişkenlere erişim sağlatılır.

Değişkenler private olarak tanımlanır.

Bu yaklaşım sadece kalıtımda değil, sınıflardan bir eleman türetildiğinde, türetilmiş objenin erişebileceği değişken ve fonksiyonları tanımlamada da kullanılır.

# Objeler ve Sınıflar

## Private fonksiyon ve değişkenler

### Örnek Kod Parçasığı

```
class C(object):  
    def __init__(self):  
        self.c = 21  
        self.__d = 42  
  
class D(C):  
    def __init__(self):  
        self.e = 84  
        C.__init__(self)  
  
object1 = D()  
print(object1.d)
```

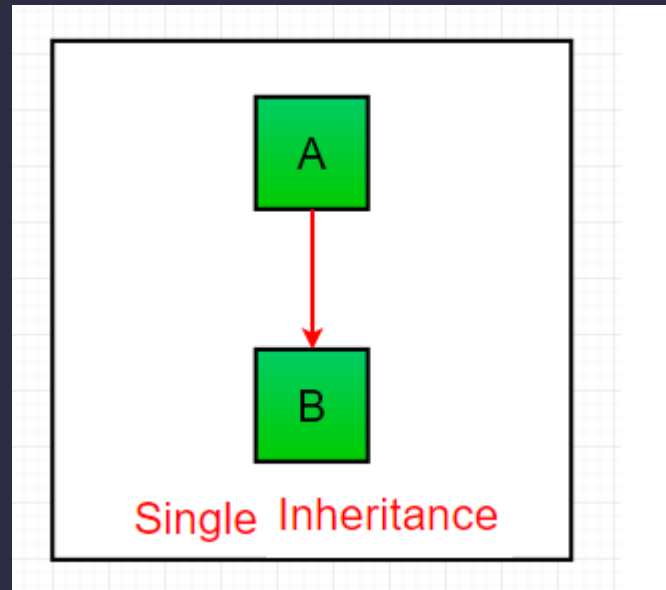
### Hata

'D' object has no attribute 'd'

# Objeler ve Sınıflar

## Örnekler

### Tekli Kalıtım



# Objeler ve Sınıflar

## Tekli Kalıtım

### Örnek Kod Parçasığı

```
class Parent:
    def func1(self):
        print("Ana sınıf fonksiyonu")

class Child(Parent):
    def func2(self):
        print("Kalitim sinifi fonksiyonu")

object = Child()
object.func1()
object.func2()
```

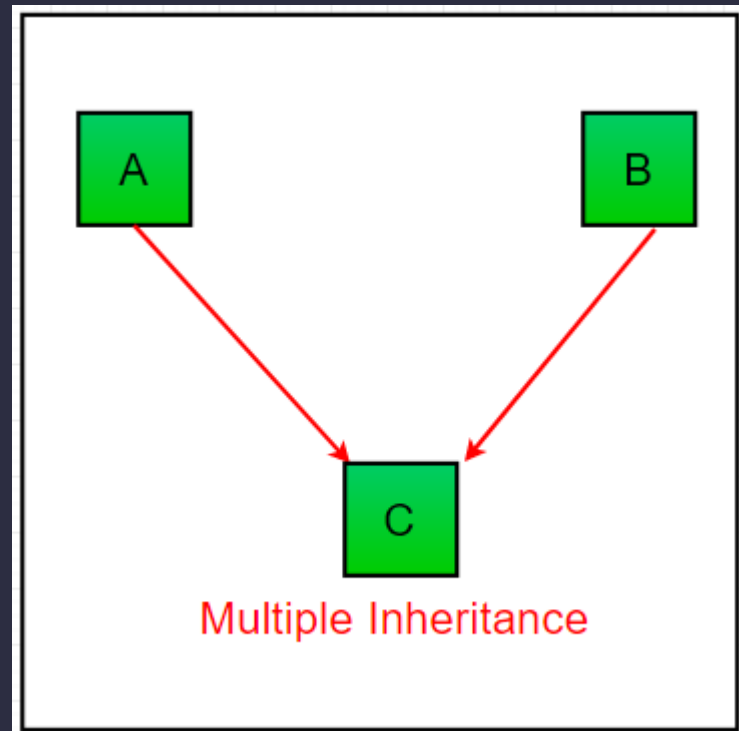
### Çıktı

Ana sınıf fonksiyonu  
Kalitim sinifi fonksiyonu

# Objeler ve Sınıflar

Örnekler

Çoklu Kalıtım



# Objeler ve Sınıflar

## Çoklu Kalıtım

### Örnek Kod Parçasığı

```
class Anne:
    anneAdi = ""
    def anneIsimAl(self):
        return self.anneAdi

class Baba:
    babaAdi = ""
    def babaIsimAl(self):
        return self.babaAdi

class cocuk(Arne, Baba):
    def aileBastir(self):
        print("Anne :", self.anneIsimAl())
        print("Baba :", self.babaIsimAl())

obj = cocuk()
obj.anneAdi = "Ayse"
obj.babaAdi = "Ahmet"
obj.aileBastir()
```

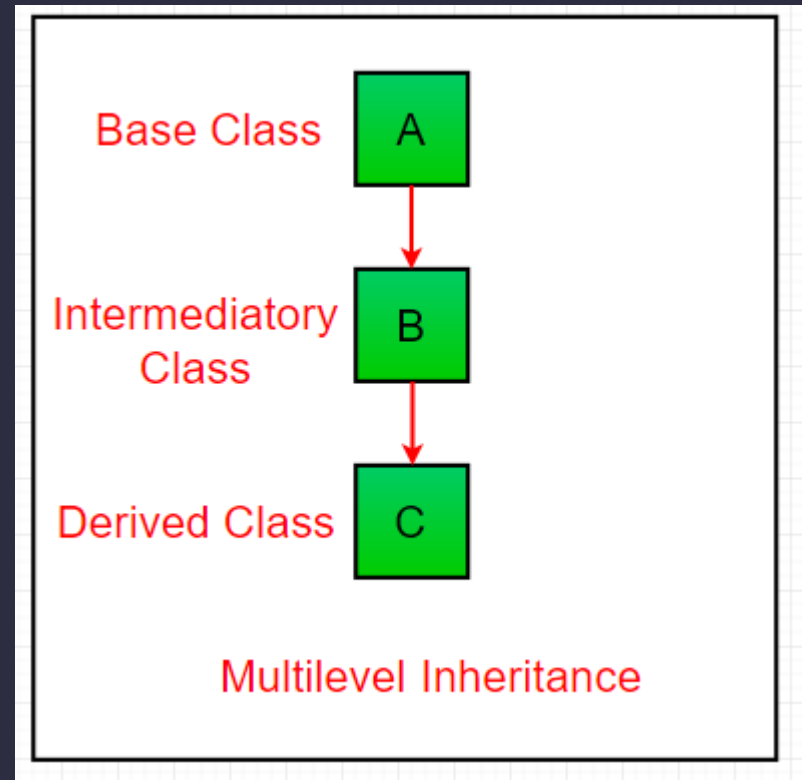
### Çıktı

Anne : Ayse  
Baba : Ahmet

# Objeler ve Sınıflar

## Örnekler

### Çoklu Seviye Kalıtım





# Objeler ve Sınıflar

## Çoklu Seviye Kalıtım

### Örnek Kod Parçasığı

```
class dede:
    dedeAdi = ""

class baba(dede):
    babaAdi = ""

class cocuk(baba):
    def aileBastir(self):
        print("Dede :", self.dedeAdi)
        print("Baba :", self.babaAdi)

obj = cocuk()
obj.dedeAdi = "Ali"
obj.babaAdi = "Mehmet"
obj.aileBastir()
```

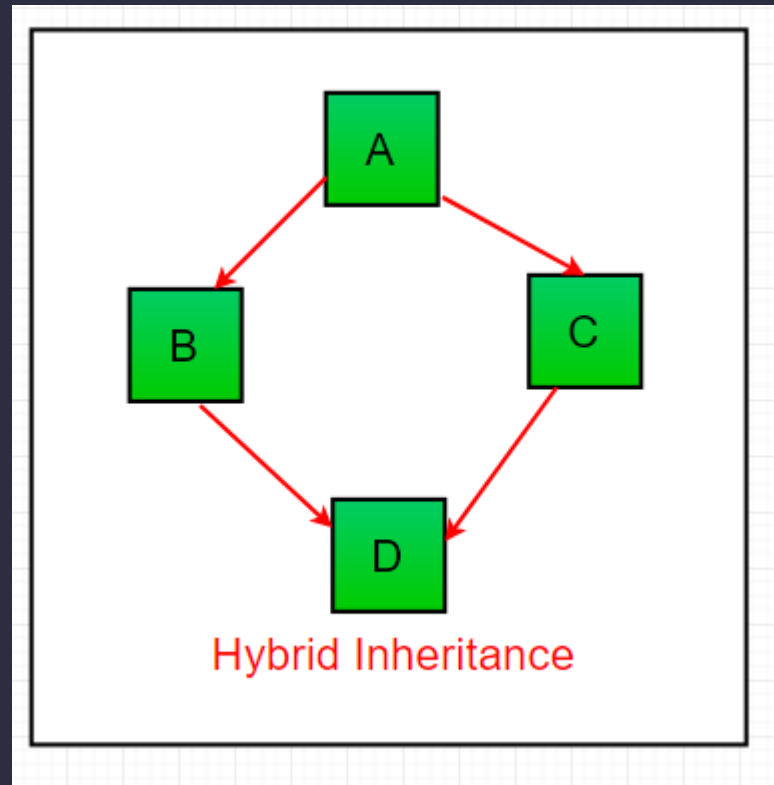
### Çıktı

Dede : Ali  
Baba : Mehmet

# Objeler ve Sınıflar

## Örnekler

### Hibrit Kalıtım



# Objeler ve Sınıflar

## Çoklu Seviye Kalıtım

### Örnek Kod Parçasığı

```
class sinif1:
    def func1(self):
        print("Fonksiyon 1")

class sinif2(sinif1):
    def func2(self):
        print("Fonksiyon 2")

class sinif3(sinif1):
    def func3(self):
        print("Fonksiyon 3")

class sinif4(sinif2, sinif3):
    def func4(self):
        print("Fonksiyon 4")

obj = sinif4()
obj.func1()
obj.func2()
obj.func3()
```

### Çıktı

Fonksiyon 1  
Fonksiyon 2  
Fonksiyon 3