

## 11. Sınıflarda arkadaşlık

Sınıflar oluşturulurken bir sınıf veya herhangi bir fonksiyon bir başka sınıfın arkadaşı olarak tanımlanabilir. Bu durumda arkadaş sınıf veya fonksiyon arkadaşı olduğu sınıfın üyelerine erişebilir. Üyelerin public, private, veya protected olması fark etmez. Aşağıda verilen programda

**DortgenKatla()** fonksiyonu Dortgen sınıfının arkadaşı olarak belirtilmiştir. Bunun için sınıf tanımı içinde bu sınıfın arkadaşı olacak fonksiyonun önüne **friend** ifadesinin getirilmesi yeterlidir.

```
#include <iostream>
#include <conio.h>
using namespace std;

class Dortgen {
    int x, y;
public:
    void DegerVer(int,int);
    int Alan (void);
    friend Dortgen DortgenKatla (Dortgen); //bu fonksiyon Dortgen sınıfının arkadaşı
};

void Dortgen::DegerVer(int a, int b) {
    x = a;
    y = b;
}

int Dortgen::Alan (void) {
    return x*y;
}

//Bu fonksiyon normalde Dortgen sınıfının uyesi değil. D1 nesnesi Dortgen sınıfından
//olduğundan bu nesneye normalde erişemez. Ancak bu fonksiyon aynı zamanda
//arkadaş olarak bildirildiğinden sınıfın bütün üyelerine erişim hakkı vardır.
Dortgen DortgenKatla (Dortgen D1)
{
    Dortgen D2;
    D2.x = D1.x*2;
    D2.y = D1.y*2;
    return (D2);
}
```

```

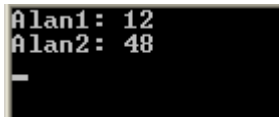
int main ()
{
    Dortgen N1,N2;

    N1.DegerVer(3,4); //N1 nesnesinin x ine 3 ve y sine 4 değeri atanır.
    N2=DortgenKatla(N1); //N2 nesnesinin x ine 7 ve y sine 2 değeri atanır.

    cout << "Alan1: " << N1.Alan()<<'\n';
    cout << "Alan2: " << N2.Alan()<<endl;

    getch();
    return 0;
}

```



```

Alan1: 12
Alan2: 48
_

```

Aşağıdaki örnekte Nesneler bir birinin katı olarak değer almakta ve ekrana yazdırılmaktadır. İnceleyiniz.

```

#include <iostream>
#include <conio.h>
using namespace std;

class Dortgen {

public:
    int x, y;
    void DegerVer(int,int);
    int Alan (void);
    friend Dortgen DortgenKatla (Dortgen); //bu fonksiyon Dortgen sınıfının arkadaşı
};

void Dortgen::DegerVer(int a, int b) {
    x = a;
    y = b;
}

int Dortgen::Alan (void) {
    return x*y;
}

```

//Bu fonksiyon normalde Dortgen sınıfının uyesi değil. D1 nesnesi Dortgen sınıfından  
//olduğundan bu nesneye normalde erişemez. Ancak bu fonksiyon aynı zamanda  
//arkadaş olarak bildirildiğinden sınıfın bütün üyelerine erişim hakkı vardır.

Dortgen DortgenKatla (Dortgen D1)

```
{  
    Dortgen D2;  
    D2.x = D1.x*2;  
    D2.y = D1.y*2;  
    return (D2);  
}
```

int main ()

```
{  
    Dortgen N1,N2;
```

N1.DegerVer(3,4); //N1 nesnesinin x ine 3 ve y sine 4 değeri atanır.

N2=DortgenKatla(N1); //N2 nesnesinin x ine 7 ve y sine 2 değeri atanır.

cout << "N1.X: " << N1.x<<endl;

cout << "N1.Y: " << N1.y<<endl;

cout << "N2.X: " << N2.x<<endl;

cout << "N2.Y: " << N2.y<<endl;

cout << "DortgenKatla(N2).X: " << DortgenKatla(N2).x<<endl;

cout << "DortgenKatla(N2).Y: " << DortgenKatla(N2).y<<endl;

cout << "N1.Alan: " << N1.Alan()<<'\n';

cout << "N2.Alan: " << N2.Alan()<<endl;

cout << "DortgenKatla(N2).Alan: " << DortgenKatla(N2).Alan()<<endl;

getch();

return 0;

```
}
```

```
N1.X: 3  
N1.Y: 4  
N2.X: 6  
N2.Y: 8  
DortgenKatla(N2).X: 12  
DortgenKatla(N2).Y: 16  
N1.Alan: 12  
N2.Alan: 48  
DortgenKatla(N2).Alan: 192  
_
```

Aşağıda verilen programda ise Bir sınıf başka bir sınıfın arkadaşı olarak tanımlanmaktadır. Dortgen sınıfı kare sınıfının arkadaşı olarak tanımlanmıştır.

Arkadaşlık ilişkisi sınıflarda tek yönlüdür. Bu Dortgen sınıfının Kare sınıfının üyelerine erişip kullanabileceğini ancak Kare sınıfının Dortgen sınıfının üyelerine erişemeyeceği anlamına gelmektedir.

```
#include <iostream>
#include <conio.h>
using namespace std;

class Kare {
    int a;
public:
    void DegerVer(int);
    friend class Dortgen; //Dortgen sınıfı Kare sınıfının arkadaşı
};

class Dortgen {
    int x, y;
public:
    void DegerVer(int,int);
    int Alan (void);
    void Donustur (Kare); //Dortgen sınıfını kareye donuşturmaktadır
};

void Kare::DegerVer(int k)
{
    a=k ;
}

void Dortgen::Donustur (Kare K)
{
    x = K.a;
    y = K.a;
}

void Dortgen::DegerVer(int a, int b) {
    x = a;
    y = b;
}

int Dortgen::Alan (void) {
    return x*y;
}
```

```
int main ()
{
    Dortgen N1;
    Kare K1;

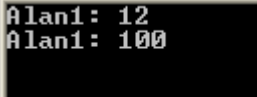
    N1.DegerVer(3,4); //N1 nesnesinin x ine 3 ve y sine 4 değeri atanır.

    cout << "Alan1: " << N1.Alan()<<'\n';

    K1.DegerVer(10);
    N1.Donustur(K1);

    cout << "Alan1: " << N1.Alan()<<'\n';

    getch();
    return 0;
}
```



```
Alan1: 12
Alan1: 100
```