BTEP243 – Ders 6

Arkadaş Fonksiyonları

Bir arkadaş fonksiyonu, bir sınıfın üyesi olmayan, ancak sınıfın "private" üyelerine erişimi olan özel bir fonksiyondur.

- Arkadaş fonksiyonlar, sınıfların dışında global olarak tanımlanmış bir fonksiyondur ve bu fonksiyon sınıf içerisindeki private verilere erişebilmeyi sağlar.
- Bildirimi, arkadaş olacağı sınıfın içinde public veya private kısmında herhangi bir yerde yapılabilir.
- Sadece sınıf ile fonksiyon arasında değil sınıf ile sınıf arasında da arkadaşlık ilişkisi kurulabilir.
- Arkadaşlık sınıf tarafından verilir ve fonksiyon veya bir başka sınıf tarafından alınır.
- Arkadaş fonksiyonu bir nesnenin "private" üyelerinde değişiklik yapacaksa sınıfın nesnesi referans parametresi olarak gönderilmelidir.
- Arkadaş fonksiyonlar simetrik (symmetric) değildir. Yani, Sınıf1, Sınıf2'nin arkadaşıysa, Sınıf2, Sınıf1'in arkadaşı olduğu anlamına gelmez.
- Arkadaşlar fonksiyonlar geçişli (transitive) değildir. Yani, Sınıf1, Sınıf2'nin arkadaşıysa ve Sınıf2, Sınıf3'ın arkadaşıysa, Sınıf1'in Sınıf3'ün arkadaşı olduğu anlamına gelmez.
- Arkadaş fonksiyonlarını kullanmak performansı arttırır.
- Arkadaş fonksiyonlarını çok dikkatli kullanmak gerekir. Yanlış arkadaş fonksiyon kullanımı, bilgi gizleme (data hiding) ve kapsülleme (encapsulation) kavramını bozabilir.

Örnek 1:

```
sınıf ornek
{
private:
        int sı;
public:
        ornek();
        friend int getnum (ornek obj);  //prototip
};
ornek :: ornek ()
{
        s = 0;
}
int getnum (ornek obj)
{
        return obj.s;
}
#include <iostream>
using namespace std;
#include "example.h"
void main ()
{
        ornek e;
        cout << getnum (e) << endl;
        system ( "pause");
}
```

Örnek2:

//Küp sınıfı ve arkadaş fonksiyon örneği

```
class kup {
private:
        int sayi;
public:
        kup()
        {
                cout << "Bir tamsayı giriniz:";
                cin >> sayi;
        }
        friend void findcube (kup &);
};
friend void kupbul (kup & obj);
{
        obj.sayi = obj.sayi * obj.sayi * obj.sayi;
        cout << obj.sayi << endl;
}

#include <iostream>
using namespace std;
#include "kup.h"
void main ()
{
        kup c;
        kupbul (c);
        cout << endl;
        system ( "pause");
}
```

Örnek 3:

```
class sayac {
private:
        int x;
public:
        sayac () { x = 0; }
        void yaz () {cout << x << endl;}
        friend void setx (sayac &, int);
};
void setx (sayac & c, int v);
{
        c.x = v;
}

#include <iostream>
using namespace std;
#include "sayac.h"
void main ()
{
        sayac sobj;
        cout << "sayac.x değeri:";
        sobj.yaz ();
```

```
	cout << "set.x arkadaş fonksiyonu çağrıldıktan sonra sayac.x:";
	setx (sobj, 8);
	sobj.yaz ();
	system ( "pause");
}
```

Çıktı:
sayac.x değeri: 0
set.x arkadaş fonksiyonu çağrıldıktan sonra sayac.x: 8

Örnek4:

```
class ogrenci {
private:
	int hnotu;
public:
	friend void goster (ogrenci &);
	void set_hnotu (int ynot)
	{
		hnotu = ynot;
	}
};
void goster (ogrenci & og)
{
  cout << og.hnotu;
}
```

```
# Include <iostream>
using namespace std;
#include "ogrenci.h"
void main ()
{
	ogrenci ali;
	ali.set_hnotu (80);
	goster (ali);
	system ( "pause");
}
```

Örnek 5: (Değer-ile-gönderme yöntemi)

```
sınıf ogrenci
{
	int ogrno;
	char hnotu;
	float cgpa;
public:
	void set ();
	friend void goster (ogrenci);
};
void ogrenci :: set ()
{
	cout << "ogrenci no, harf notu ve cgpa giriniz:";
```

```
	cin >> ogrno >> hnotu >> cgpa;
}
void goster (ogrenci p)
{
	cout << p.ogrno << " " << p.hnotu << "" << p.cgpa;
}

#include <iostream>
using namespace std;
#include "ogrenci.h"
void main ()
{
	ogrenci obj;
	obj.set ();
	goster (obj);
	system ( "pause");
}
```

Örnek 6: (Referans-ile-gönderme yöntemi)

```
sınıf ogrenci
{
	int ogrno;
	char hnotu;
	float cgpa;
public:
	friend void set (ogrenci &);
	void goster ();
};
void ogrenci :: set (ogrenci &obj)
{
	cout << "ogrenci no, harf notu ve cgpa giriniz:";
	cin >> obj.ogrno >> obj.hnotu >>obj.cgpa;
}
void ogrenci::goster ()
{
	cout << ogrno << " " << hnotu << "" << cgpa;
}

#include <iostream>
using namespace std;
#include "ogrenci.h"
void main ()
{
	ogrenci obj;
	set (obj);
	obj.goster ();
	system ( "pause");
}
```

Örnek 7: **(Adres-ile-gönderme yöntemi)**

```
sınıf ogrenci
{
        int ogrno;
        char hnotu;
        float cgpa;
public:
        friend void set (ogrenci *);
        void goster();
};
void set (ogrenci * obj)
{
        cout << "ogrenci no, harf notu ve cgpa giriniz:";
        Cin >> obj->ogrno >> obj->hnotu >> obj->cgpa;
}
void ogrenci::goster()
{
        cout << ogrno << "" << hnotu << "" << cgpa;
}

#include <iostream>
using namespace std;
#include "ogrenci.h"
void main ()
{
        ogrenci obj;
        set(&obj);
        obj.goster ();
        system ( "pause");
}
```

Örnek 8:

```
class silver {
        int x, y;
private:
silver()
{
        x = 1; y = 2;
}
void goster ()
{
        cout << x << " " << y << endl;
}
void reset ()
{
        x- = 2;
        y * = 2;
}
friend void fonk1 (silver);
friend void fonk2 (silver &);
};
```

```
void fonk1 (silver s)
{
	cout << s.x << " " << s.y << endl;
	s.x + = 4;
	s.y + = 2;
	cout << s.x << " " << s.y << endl;
}
void fonk2 (silver& s)
{
	cout << s.x << " " << s.y << endl;
	s.x + = 2;
	s.y + = 3;
}
#include <iostream>
using namespace std;
#include "silver.h"
void main ()
{
	silver s1;
	fonk1 (s1);
	s1.reset ();
	fonk2 (s1);
	s1.goster ();
	system ( "pause");
}
```

Örnek 9: İKİ SINIF İLE ARKADAŞ OLAN FONKSİYON ÖRNEĞİ

```
class beta; // ileri beyan
class alfa {
private:
	int x;
public:
	void getx ()
	{
		cout << " x değerini giriniz:";
		cin >> x;
	}
	friend void karsilastir (alfa, beta);
};
class beta {
private:
	int y;
public:
	void getxy ()
	{
		cout << " y değerini giriniz:";
		cin >> y;
	}
	friend void karsilastir (alfa, beta);
};
```

```
void karsilastir (alfa a, beta b)
{
	if (a.x> b.y)
		cout << "Buyuk deger:" << a.x << endl;
	else if (b.y> a.x)
		cout << "Buyuk deger:" << b.y << endl;
	else
		cout << "Sayilar birbirine esit" << endl;
 }
#include <iostream>
using namespace std;
#include "ab.h"
void main ()
{
	alfa obj1;
	beta obj2;
	obj1.getx ();
	obj2.gety ();
	karsilastir (obj1, obj2);
	system ( "pause");
}
```

Çalışma Soruları

Aşağıdaki programların ekran çıktısını bulunuz.

a)

```
class test {
          int x;    double y;
public:
  test(int=5);
  void set(int);
  int get(int);
  double sqr();
  void f1(int&t,double&d1);
  void show();
  friend void f2(test&m);
};
test::  test  (int k)
{  x=k ; y=2.0*k;          }

void test :: set  (int t)
{ x=t;t=t*t; y=t;  }

int test :: get ( int k )
{
if (k==1) return x;
else return x*x;          }

double test :: sqr ()
{ x=x*x ; y=y*y;
          return x*y;      }

void test :: f1 (int&t, double&d1)
{ t=x  ; d1=y;  }
void test::show()
{ cout<<"\nX = "<<x<<"   Y= "<<y<<endl;  }

void  f2 (test&m)
{
m.x+=5 ; m.y*=2.0;
}
```

```
#include <iostream>
using namespace std;
#include "test.h"
void main()
{
int a; double b;
test obj1, obj2(3);
obj1.show();
obj2.show();
obj1.set(2);
obj1.show();
cout<<"\n"<<obj2.sqr();
obj2.show();
obj1.f1(a,b);
cout<<"\n"<<a<<"  "<<b;
obj2.f1(a,b);
cout<<"\n"<<a<<"  "<<b;
f2(obj1);
f2(obj2);
obj1.show();
obj2.show();
}
```

b)

```
class test {
        int a , b ;
public:
 test ( int  x = 2 , int y = 1 )
 { a = x * 3 ; b = y * 2 ; }
 void show()
 { cout<<a<<"  "<<b<<endl; }
 friend void f1 ( test & );
 int f2 ( int  c , int  & d )
 {
        a = c + 1 ; b  = d + 2 ; c = 4 ; d = 3 ;
        return ( a*c - b*d) ;
}
};

void f1 ( test & s )
{
        cout<<s.a<<"  "<<s.b<<endl;
         s.a=3 ; s.b = 5 ;
}
```

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

void main()
 {
 test t1, t2(5) , t3(3,2) ;
 t1.show();
 t2.show();
 t3.show();
 int k = 2 , m = 5 ;
 cout<<t2.f2(k,m)<<endl;
 cout<<k<<"  "<<m<<endl;
 t2.show();
 f1(t3);
 t3.show();
 }
```