

LECTURE-9

OBJECT ORIENTED PROGRAMMING C++

06/09/2020

Dr. Farhana Sarker
Assistant Professor
CSE, ULAB

FRIEND FUNCTION

A friend function is a function which can be given special grant to access **all the members (private, protected and public)** of a class where it is declared as friend.

A friend function can be:

- a) A method of another class
- b) A global function

ADVANTAGES OF FRIEND FUNCTION

- Friend functions can access **all the members (private, protected and public)** of a class where it is declared as friend.

SYNTAX OF FRIEND FUNCTION

```
class class_name
```

```
{
```

Access specifier:

```
friend return type function
```

```
name(class_name &object name,
```

```
int i, char j);
```

```
}
```

```
};
```

return type function

```
name(class_name
```

```
&object name, int i,  
char j)
```

```
{
```

Body of the function;

```
}
```

EXAMPLE |

```
class A
{
private:
int a,b,c;
protected:
void output();
friend void hello(A &obj);
};
```

```
void hello(A &obj)
{
    obj.output();
}
```

```
int main()
{
A on;
hello(on);
return 0;
}
```

EXAMPLE 2

```
class A
{
int a,b,c;
protected:
void output()
{
cin>>a>>b>>c;
cout<<a<<b<<c;
}
friend void hello(A &obj);
};
```

```
void hello(A &obj)
{
int c,d,a;
cin>>obj.a>>obj.b>>obj.c;
cin>>c>>a>>d;
cout<<obj.a+obj.b+c+d+obj.c;
}
```

```
int main()
{
A o;
hello(o);
return 0;
}
```

SOME IMPORTANT POINTS TO NOTE

- Friends should be used only for limited purpose, too many functions or external classes are declared as friends of a class with protected or private data, it lessens the value of encapsulation of separate classes in object-oriented programming.
- Friendship is not mutual. If class A is a friend of B, then B doesn't become a friend of A automatically.
- Friend function of a class is defined outside that class's scope
- Friendship is not **inherited**. (we will understand this in our inheritance lecture)

FRIEND CLASS

A friend class can access **all the members (private, protected and public)** of other class in which it is declared as friend.

SYNTAX OF FRIEND CLASS

```
class class_name1  
{  
Access specifier:  
friend class_name2;  
}  
};
```

```
class class_name2{  
Body of the class2;  
}
```

EXAMPLE |

```
class A
{
private:
int a,b,c;
protected:
void output()
{
cin>>a>>b>>c;
cout<<a<<b<<c;
}
friend class B;
};
```

```
class B{
public:
void hello(A &obj)
{
obj.output();
}
private:
char p;
};
```

```
int main()
{
A on;
B fs;
fs.hello(on);
return 0;
}
```