

Object Oriented Programming

Lecture



Friend Functions and Classes





Remember:

- Private members of a class cannot be accessed from outside the same class in which they are declared

However:

- This rule does not apply to "*friends*"

Definition

*“A function outside of a class can be defined to be a **friend function** by the class which gives the friend function **free access** to the **private (or protected)** members of the class”*



Friends Function

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
public:
    void set_values(int, int);
};
```



Friends Function

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
public:
    void set_values(int, int);

};

int area(Rectangle a) {
    return a.width * a.height; //Error
}
```



Friends Function

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
public:
    void set_values(int, int);
    friend int area(Rectangle a);
};

int area(Rectangle a) {
    return a.width * a.height;
}
```

Now the function **Area** can access data members of class **Rectangle**

Q) Why Not Just Use a Normal Function?

- 1) A normal function cannot access private or protected members of a class unless you use getter functions.
- 2) A friend function can directly access private members, even though it is not part of the class.



Friends Function

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
public:
    void set_values(int, int);
    friend int area(Rectangle a);
};

int area(Rectangle a) {
    return a.width * a.height;
}

void Rectangle::set_values(int x, int y) {
    width = x;
    height = y;
}
```



Friends Function

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
public:
    void set_values(int, int);
    friend int area(Rectangle a);
};
```

```
int area(Rectangle a) {
    return a.width * a.height;
}
```

```
void Rectangle::set_values(int x, int y) {
    width = x;
    height = y;
}
```

```
int main() {
    Rectangle rect;
    rect.set_values(3, 4);
    cout << "area: " << area(rect);
    return 0;
}
```

friend keyword is not
given in definition



Friends Function

- Friend functions can be placed anywhere in the class without any effect
- Access specifiers don't affect friend functions or classes

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
public:
    void set_values(int, int);
    friend int area(Rectangle); ← Public
};
```

```
#include <iostream>
#include <string>
using namespace std;

class Rectangle {
    int width, height;
    friend int area(Rectangle); ← Private
public:
    void set_values(int, int);
};
```



Friends Classes

Similarly, one class can also be made friend of another class:

```
class Square {  
    int width, height;  
public:  
    void set_values(int x, int y);  
    friend class Rectangle;  
};  
  
void Square::set_values(int x, int y) {  
    width = x;  
    height = y;  
}
```



Friends Classes

```
class Square {  
    int width, height;  
public:  
    void set_values(int x, int y);  
    friend class Rectangle;  
};  
  
void Square::set_values(int x, int y) {  
    width = x;  
    height = y;  
}
```

```
class Rectangle {  
    int width, height;  
public:  
    void set_values(Square x);  
  
};  
  
void Rectangle::set_values(Square x) {  
    width = x.width;  
    height = x.height;  
}
```



Friends Classes

```
class Square {  
    int width, height;  
public:  
    void set_values(int x, int y);  
    friend class Rectangle;  
};  
  
void Square::set_values(int x, int y) {  
    width = x;  
    height = y;  
}
```

```
class Rectangle {  
    int width, height;  
public:  
    void set_values(Square);  
};  
  
void Rectangle::set_values(Square x) {  
    width = x.width;  
    height = x.height;  
}  
  
int main() {  
    Square sq;  
    sq.set_values(3, 4);  
    Rectangle rect;  
    rect.set_values(sq);  
    return 0;  
}
```

Important thing to remember:

- “Friend” opens a **small hole in the protective shield** of the class, so it should be used very carefully
- You should implement this only when there is no way to solve your programming problem



Thanks a lot

