Data Structures and Object Oriented Programming

Lecture 4

Dr. Naveed Anwar Bhatti

Webpage: naveedanwarbhatti.github.io

Object-Oriented Programming in C++

Friendship and inheritance

Friends Function

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;
}
```

Friends Function

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

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

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

Friends Function

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

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

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);
};

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;
}
```

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;
}
```

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;
}
```

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;
}
```

Friend Functions and Classes

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

Friend Functions and Classes

Exercise:

Consider these two classes:

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

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

Write a friend function '**add**' which add the objects of these two classes and print them.

Friend Function

```
class Rectangle;

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

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

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

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

void add(Square A, Rectangle B)
{
cout << "Width = "<< A.width + B.width<< endl;
cout << "Height = " << A.height + B.height <<
endl;
}

int main() {
    Square s;
    Rectangle r;
    s.set_values(1, 1);
    r.set_values(1, 1);
    add(s, r);
    return 0;
}
```