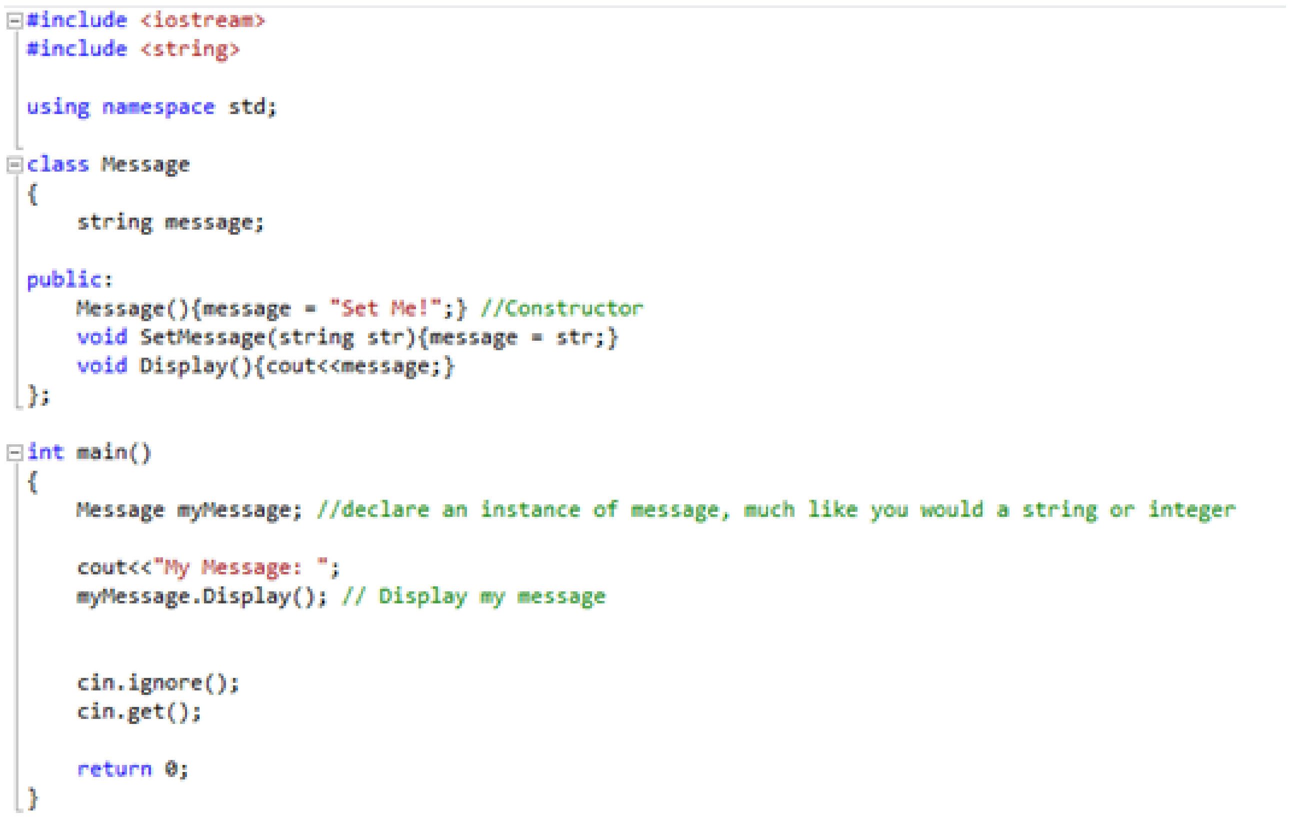
Lab 9 - wb 13/01/14

This week we are looking more at classes. We will look at constructors and then at some basic inheritance.

Constructors

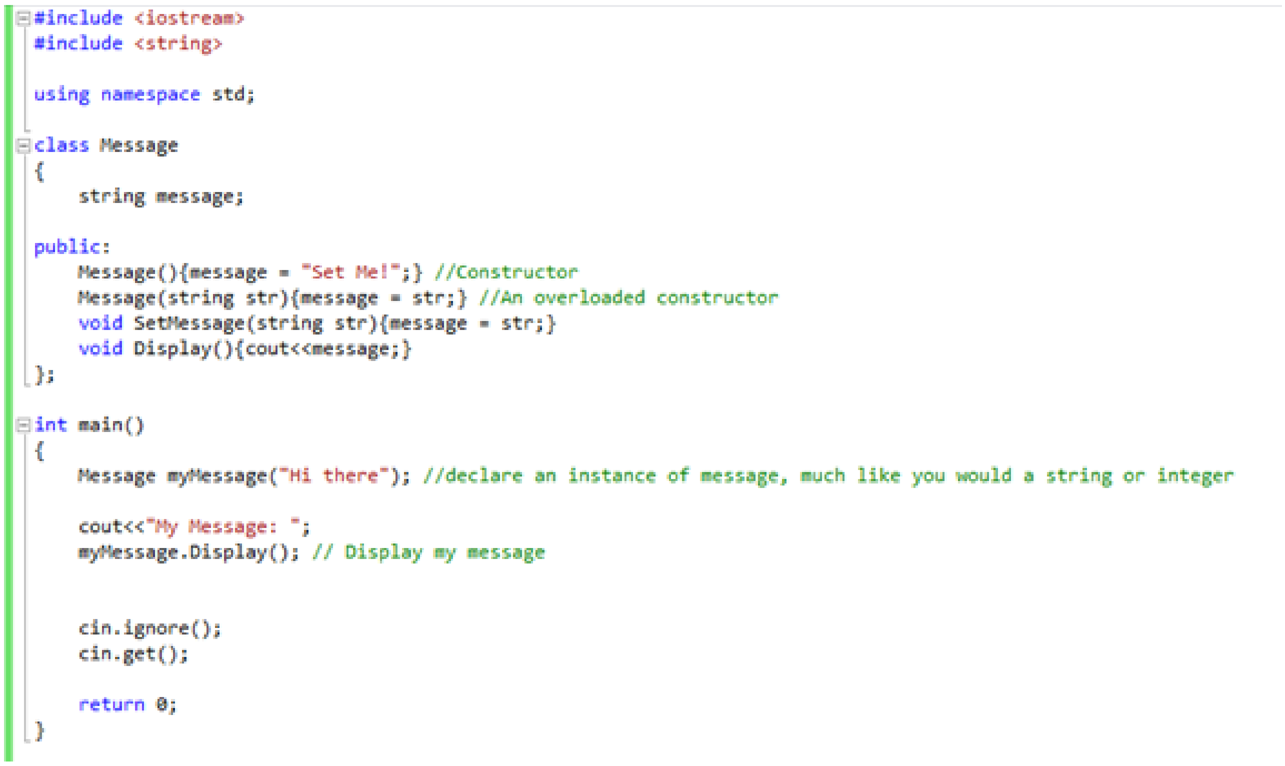
We will start with last weeks classes demo.



```
#include <iostream>
#include <string>

using namespace std;

class Message
{
    string message;

public:
    Message(){message = "Set Me!";} //Constructor
    void SetMessage(string str){message = str;}
    void Display(){cout<<message;}
};

int main()
{
    Message myMessage; //declare an instance of message, much like you would a string or integer

    cout<<"My Message: ";
    myMessage.Display(); // Display my message


    cin.ignore();
    cin.get();

    return 0;
}
```

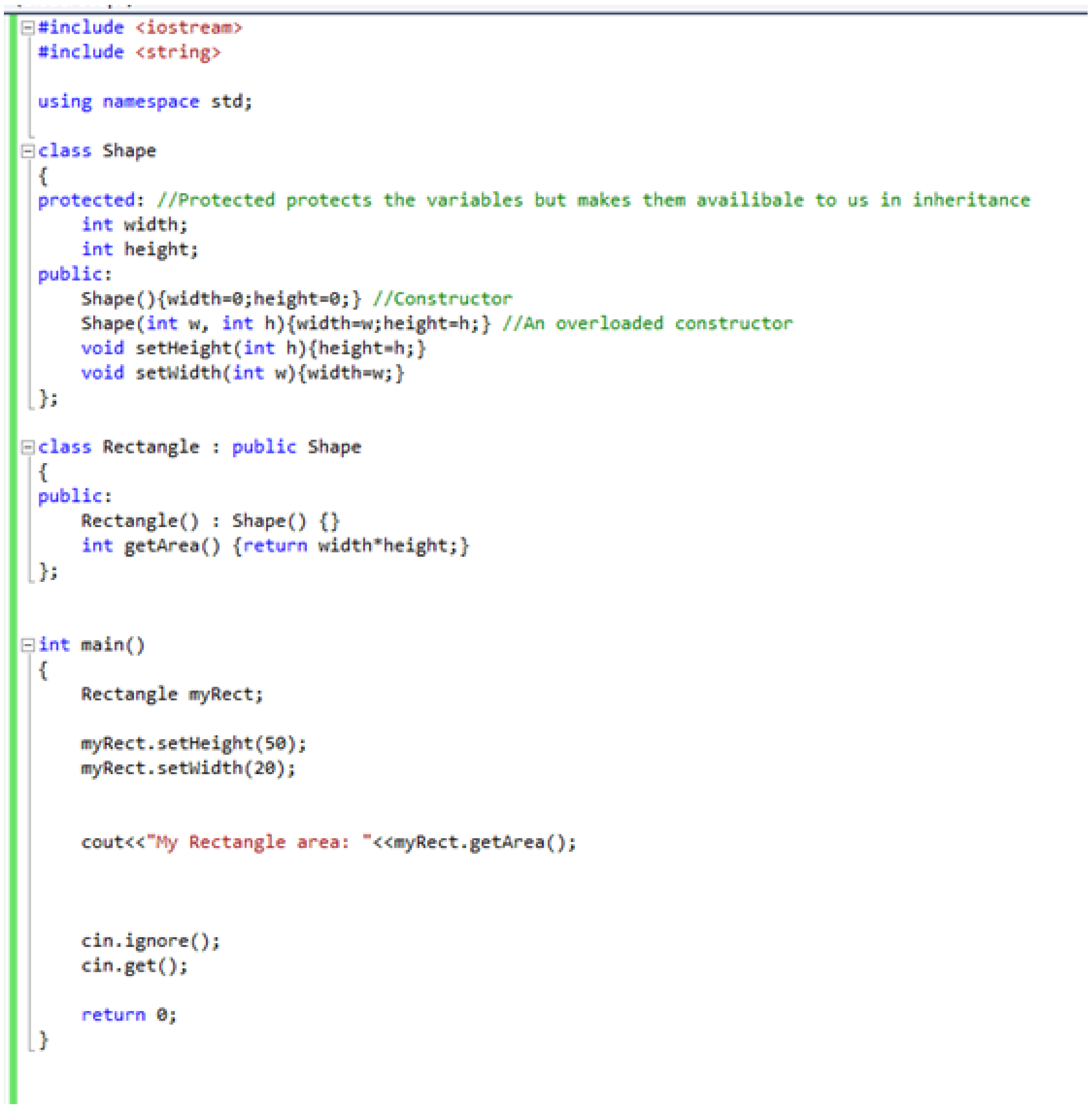
Overloaded Constructors

With overloaded constructors we can set variables at the same time as declaring our instance of the class.



```
#include <iostream>
#include <string>

using namespace std;

class Message
{
    string message;

public:
    Message(){message = "Set Me!";} //Constructor
    Message(string str){message = str;} //An overloaded constructor
    void SetMessage(string str){message = str;}
    void Display(){cout<<message;}
};

int main()
{
    Message myMessage("Hi there"); //declare an instance of message, much like you would a string or integer

    cout<<"My Message: ";
    myMessage.Display(); // Display my message


    cin.ignore();
    cin.get();

    return 0;
}
```

Inheritance
Inheritance allows us to reuse another class to make one thats similar.



```
#include <iostream>
#include <string>

using namespace std;

class Shape
{
protected: //Protected protects the variables but makes them availibale to us in inheritance
    int width;
    int height;
public:
    Shape(){width=0;height=0;} //Constructor
    Shape(int w, int h){width=w;height=h;} //An overloaded constructor
    void setHeight(int h){height=h;}
    void setWidth(int w){width=w;}
};

class Rectangle : public Shape
{
public:
    Rectangle() : Shape() {}
    int getArea() {return width*height;}
};


int main()
{
    Rectangle myRect;

    myRect.setHeight(50);
    myRect.setWidth(20);


    cout<<"My Rectangle area: "<<myRect.getArea();



    cin.ignore();
    cin.get();

    return 0;
}
```

Have A Go - Test Your Knowledge!

Challenge 1

Alter the inheritance demo to add another shape and give it’s area. Such as a triangle.