C++ OOP Header Files Rectangle Tutorial

Contents

Time required: 60 minutes

Step 1: Single File Rectangle OOP

We will start with a C++ OOP program in a single file.

Enter the following code. This code defines the Rectangle class.

```cpp
/**
 * Filename: rectangle_oop.cpp
 * Written by:
 * Written on:
 * Revised:
 * Calculate the perimeter and area of a rectangle with C++ OOP
 */
#include <iostream>

//*************************************************************************
//  Define Rectangle Class
//*************************************************************************
class Rectangle
{
    // ------------------ DEFINE PUBLIC METHODS ---------------- //
public:
    // 2 parameter constructor
    Rectangle(double width, double length)
    {
        this->m_width = width;
        this->m_length = length;
    }

    double get_perimeter()
    {
        return ((this->m_width * 2) + (this->m_length * 2));
    }

    double get_area()
    {
        this->m_area = this->m_width * this->m_length;
        return m_area;
    }

    // -------------- DEFINE PRIVATE DATA MEMBERS -------------- //
private:
    double m_width;
    double m_length;
    double m_area;
};
```

```
42  //************************************************************************
43  //  main program entry point
44  //************************************************************************
45  int main()
46  {
47      double width{2.5};
48      double length{7.0};
49      std::cout << "\n -- Rectangle Calculator Single File --\n" << std::endl;
50      // Create rectangle object with 2 arguments
51      Rectangle rectangle1(width, length);
52
53      // Return and display values from object methods
54      std::cout << " Perimeter: " << rectangle1.get_perimeter() << std::endl;
55      std::cout << " Area: " << rectangle1.get_area() << std::endl;
56      std::cout << std::endl;
57
58      // Pause until a key is pressed
59      system("PAUSE");
60      return 0;
61  }
```

Example run:

```
 -- Rectangle Calculator Single File --

 Perimeter: 19
 Area: 17.5

Press any key to continue . . .
```

Step 2: Rectangle OOP with Multiple Files

As projects become bigger, placing all the code in a single file becomes unmanageable.

rectangle.h defines the public method prototypes and private data members.

```h
/**
 * Filename: rectangle.h
 * Written by:
 * Written on:
 * Revised:
 * Header file which defines prototypes for Rectangle class
 */

#ifndef RECTANGLE_H
#define RECTANGLE_H

//*************************************************************************
// Define Rectangle Class
//*************************************************************************
class Rectangle
{
    // -------- DEFINE PUBLIC METHOD PROTOTYPES ---------------- //
public:
    // 2 parameter constructor prototype
    Rectangle(double width, double length);
    double get_perimeter();
    double get_area();

    // -------------- DEFINE PRIVATE DATA MEMBERS -------------- //
private:
    double m_width;
    double m_length;
    double m_area;
};
#endif
```

rectangle.ccp implements the public methods and data members defined in the **rectangle.h** file.

```cpp
/**
 * Filename: rectangle.cpp
 * Written by:
 * Written on:
 * Revised:
 * Implement Rectangle class constructor and methods
 */

#include "rectangle.h"

// ------------------ IMPLEMENT PUBLIC METHODS -----------------//
// Implement 2 parameter constructor
Rectangle::Rectangle(double width, double length)
{
    this->m_width = width;
    this->m_length = length;
}

double Rectangle::get_perimeter()
{
    return ((this->m_width * 2) + (this->m_length * 2));
}

double Rectangle::get_area()
{
    this->m_area = this->m_width * this->m_length;
    return m_area;
}
```

rectangle_main.cpp is the main program which uses the previous 2 files.

```cpp
/**
 * Filename: rectangle_main.cpp
 * Written by:
 * Written on:
 * Revised:
 * Calculate the perimeter and area of a rectangle
 * using header files with OOP
 */
#include <iostream>
#include "rectangle.h"

int main()
{
    double width{2.5};
    double length{7.0};
    std::cout << "\n -- Rectangle Calculator Multiple Files --\n" << std::endl;
    // Create rectangle object with 2 arguments
    Rectangle rectangle1(width, length);

    // Return values from object methods
    std::cout << " Perimeter: " << rectangle1.get_perimeter() << std::endl;
    std::cout << " Area: " << rectangle1.get_area() << std::endl;
    std::cout << std::endl;

    // Pause until a key is pressed
    system("PAUSE");
    return 0;
}
```

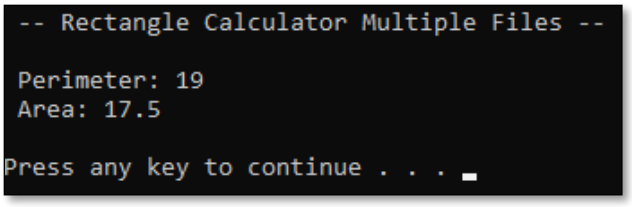
Step 3: Build G++ Multiple File Projects

```
# -Wall is optional, it shows all compile warnings
g++ -Wall rectangle.h rectangle_main.cpp rectangle.cpp -o rectangle_main.exe
pause
```

Instead of typing this in every time, let's make a batch file for this.

1. In VSCode → create a file named: **compile.bat**
2. Copy the above code and paste it into the batch file.
3. The pause command on the second line pauses the command line window so you can see if it worked or not.
4. In File Explorer → Double Click **compile.bat**
   NOTE: When compiling from the command line, you will not get any feedback unless there is an error. The command will return to a command prompt

Example program run:



Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.