

CSC2111

Computer Science I

Lab 12

50 points

Due 03/05/16 10:00 pm

Use comments in your code (5 points). Your solution must include header, implementation file, and test files (5 points).

Consider a graphics system that has classes for various figures rectangles, squares, triangles, circles, and so on. For example, a rectangle might have data members for Height, Width and center point, while a square and circle might have only a center point and an edge length or radius.

In a well-designed system, these would be derived from a common class, **Figure**. You are to implement such a system.

The class Figure is the base class. You should add only **Rectangle** and **Triangle** classes derived from Figure.

Each class has two member functions

- ✓ erase
- ✓ draw

Each of the member functions **erase** and **draw** outputs a message telling what function has been called and what the class of the calling object is. For example if you call the function erase from the object of type Rectangle, the output is "**this is the function Rectangle::erase()**". (These functions do nothing more than output this message.)

The base class has another extra function named **center**. The member function center will also output a message like this: "**the member function center is called to erase and draw**". Then it will call the erase and draw functions to erase and draw the figure. So basically this function just write a line in the output and then it will call erase and draw functions.

All the member functions should take no arguments.

There are three parts to this project:

- ✓ Write the class definitions using no virtual functions. Compile and test. (15 points)
- ✓ Make the base class member functions virtual. Compile and test (15 points).
- ✓ Explain the difference in results (10 points).

Use the following main function for all testing (**Remember to include necessary files before main**):

```
int main()
{
    Triangle tri;
    tri.draw();
    cout << "\n In main, Derived class Triangle object calling"
          << " center().\n";
    tri.center();
    Rectangle rect;
    rect.draw();
    cout << "\n In main, Derived class Rectangle object calling"
          << " center().\n";
    rect.center();
    return 0;
}
```