Complete the following program with the given declarations. All user input and output **must** occur in your `main` function. You are permitted to create other functions if you feel they would help you complete the task. Call your C++ file Shapes. This is to be submitted to Classroom by the date on the assignment. You are only completing one C++ file for this mini-project.

In your `main` function, display a menu to ask the user for their desired task. They are to first choose their operation and then enter the values required. Their options are listed below. When the process is complete, ask the user if they want to do it again.

Create a class called `Polygon` inside of your program. Your class must contain protected attributes for the length and width for a given `Polygon` object. You must also create a constructor to set the length and width for your object. Your class must also contain the following methods, as well as any relevant accessor and mutator methods.

In class `Polygon`:

`void setValues(int a, int b)`

This method sets both the length and width for the `Polygon`.

You must also create at least <u>two other</u> derived classes that deal with specific types of `Polygon`s, such as `Rectangle` or `Triangle`. These classes must have the following methods:

`float area()`

This method returns the area of the `Polygon`.

In your `main` function, you must allow the user to pick their choice of `Polygon` and enter its information. You must show the invocation using both the constructor and the `setValues` method, although either one may be commented out if you choose to do so. After the user has entered the information, display the relevant information about the `Polygon` using the `Output` class.

The `Output` class must contain the following public method:

`void print();`

This method prints the length, width, and area of the calling object.

As a hint, this project uses multiple inheritance (YAY C++!).

This project will be graded on:

1. Documentation (4 points): Your code must be fully commented and employ standard C++-style conventions.
2. User-Friendliness (4 points): All interactions with the user must be clear and unambiguous.
3. Classes (24 points): Each function has the proper declaration and works as expected.

TOTAL: 32 points