Triangle Class

For this lab, you will create a simple class that creates and evaluates triangles.

Program Design

You will create a class and then use the provided test program to make sure it works. This means that your class and methods must match the names used in the test program.

Also, you need to break your class implementation into two files. Triangle.h should include the class declaration and Triangle.cpp should include the function or method definitions. One way to do this is to get everything working in main.cpp and then create the new files and move the appropriate code to them. For Code::Blocks, you want to use the file/new/Class tab to create your class. There are examples in Moodle for your reference. This is similar to what we did last term when we split a function into a header and an implementation file.

Remember that when writing a method outside of the class, you need to specify the class name in the method name, for example to have a validate method for triangle, you would use:

```
bool Triangle::validate()
{
  // do something
}
```

When all is working, you should zip up your complete project and submit it.

Program Requirements

You need to define a class to implement a triangle.

1. The triangle is defined by the length of its 3 sides. Each length is an integer. You can name the sides whatever you want, but you do need three variables.

2. The triangle must have a default constructor that sets the length of each side to 3, 4 and 5 respectively. The triangle must also have an overloaded construction that takes three arguments so that the length of each side can be specified.

3. The triangle must have three getter (accessor) methods (getA(), getB(), getC()) that allow a programmer to retrieve the length of each side. These methods must have these names so they will work with the test function, but you can internally use any name you want for the sides.

4. The triangle must have three setter (mutator) methods (setA(int), setB(int), setC(int)) that allow a programmer to change the length of each side.

5. The triangle must have four additional methods isEquilateral(), isScalene(), isIsosceles(), and isRight(). Each of these last four methods has no parameters and returns a boolean value that is true if the appropriate condition is met.

Triangle Class

Program Hints

If you are not familiar with the terms used in the program requirements:

A scalene triangle is one where no two sides are equal in length

An isosceles triangle is one where at least two sides are equal in length

An equilateral triangle is one where all three sides have the same length

A right triangle meets the Pythagorean Theorem where the square of the length of one side is equal to the sum of the squares of the lengths of the other two sides. You need to test all three possibilities or you need to determine ahead of time which one is the longest. Your choice how you want to do this.