

CIS 230 – Spring 2006
Homework #5
Due Wednesday, March 22, 2006 at 10:00 AM

Submit your files using the submission tool on the qs-server:
/class-files/gilden/230submit

Files to Submit:

- triangle.h (unmodified)
- triangle.cpp (with your modifications)
- hw6.cpp (unmodified)

The files triangle.h, triangle.cpp and hw6.cpp are in my class-files directory on qs-server.

You may copy them to your current working directory by typing:

```
>cp /class-files/gilden/hw06/* .
```

(note the “.” at the end (after a space)– this tells the copy command to place all of the files you YOUR current working directory)

Do not modify the triangle.h file!! Do not modify hw6.cpp.

Assignment:

The file *triangle.h* contains the declaration for the class *Triangle*:

```
class Triangle
{
    private:

        float side1, side2, side3;

    public:

        Triangle(float s1, float s2, float s3);
        // constructor for Triangle object

        Triangle();
        // if no arguments, set each side's length to zero

        void sort();
        //sort the sides of the Triangle

        void swap(float &, float &);
        //swap two sides

        void getSides(float &, float &, float &);
        //returns the sides of the Triangle

        float calcPerimeter();
        //returns the perimeter of the Triangle

        float calcArea();
        // returns the area of the Triangle.  See notes on Heron's Formula

        char isTriangle();
        //returns 'Y' if three sides make a Triangle; 'N' otherwise
```

```

    char isIsosceles();
    // returns 'Y' if Triangle is isosceles; 'N' otherwise

    char isEquilateral();
    // returns 'Y' if Triangle is equilateral; 'N' otherwise

    char isRight();
    // returns 'Y' if Triangle is a right Triangle; 'N' otherwise

    char isObtuse();
    // returns 'Y' if Triangle is an obtuse Triangle; 'N' otherwise
};

```

The file *hw6.cpp* contains several invocations upon member functions for *Triangle* objects:

```

#include <iostream>
#include "triangle.h"
using namespace std;

int main()
{
    float sideOne, sideTwo, sideThree;

    cout << "Enter the values for three sides now: " << endl;
    cin >> sideOne >> sideTwo >> sideThree;

    Triangle myFigure(sideOne, sideTwo, sideThree);    //call Triangle constructor function
    if ( myFigure.isTriangle() == 'Y')                //call Triangle isTriangle function
    {
        myFigure.getSides(sideOne, sideTwo, sideThree);
        cout << "A triangle with sides: " << sideOne << ", " << sideTwo << ", " << sideThree << endl;
        cout << "has a perimeter of: " << myFigure.calcPerimeter() << endl;
        cout << "and an area of: " << myFigure.calcArea() << endl;
        cout << "Is this triangle an isosceles triangle? "<< myFigure.isIsosceles() <<endl;
        cout << "Is this triangle an equilateral triangle? " << myFigure.isEquilateral() << endl;
        cout << "Is this triangle a right triangle? " << myFigure.isRight() << endl;
        cout << "Is this triangle an obtuse triangle? " << myFigure.isObtuse() << endl;
    }
    else
        cout << "These sides do not construct a triangle." << endl;

    return 0;
}

```

The file *triangle.cpp* contains the beginnings of the class definition. It has the constructor functions defined, as well as all the member function headers. However, it leaves the member function bodies for you to write!

Your job is to write the Triangle class member functions `isTriangle()`, `getSides()`, `calcPerimeter()`, `calcArea()`, `isIsosceles()`, `isEquilateral()`, `isRight()` and `isObtuse()`.

Calculating the Area:

The Area of a triangle can be calculated using Heron's Formula:

$$s = \frac{(a+b+c)}{2}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

(Note: when calculating “s” $a+b+c$ should look familiar (i.e., you already have a function that performs that calculation))

Remember that an isosceles triangle is one which has at least two sides equal; an equilateral triangle is one which has all three sides equal; a right triangle is one in which the sum of the squares of the two shorter sides equal the square upon the longest side (the hypotenuse); an obtuse triangle is one in which sum of the squares of the two shorter sides is smaller than the square upon the longest side; and that in order to be a triangle, no two sides can when added together be equal to or shorter than the third side.

For simplicity's sake, the constructor will sort the sides so that the Side 1 will be \leq Side 2 which will be \leq Side 3. That is, they will be ordered from smallest to largest. This means you don't have to find the longest side; you know it's Side 3.

Testing:

Be sure to use values that test all aspects of your program. I will be running values to test each possibility.

Some sample data you may use:

16	14	10.5
6	7.5	9
4	3	5

others??