Student name :	Carter Hawks
Student email :	ckh170000@utdallas.edu
Class name :	2336.001_F18
Submitted on :	Oct 05, 2018 09:56 pm

Driver.cpp

```
#include <iostream>
#include "Triangle.cpp"
using namespace std;

int main() {

                        double side1 = 0;
                double side2 = 0;
                double side3 = 0;
                string color = "";
                string filled = "";

                cin >> side1;
                cin >> side2;
                cin >> side3;
                cin >> color;
                cin >> filled;



                Triangle triangle(side1, side2, side3);
                triangle.setColor(color);
                if(filled.compare("true")==0)
                        triangle.setFilled(1);
                else
                        triangle.setFilled(0);
                printf("%.2f",triangle.getArea());
                cout << endl;
                printf("%.2f",triangle.getPerimeter());
                cout << endl;
                cout << triangle.toString();
                cout << endl;

                return 0;


}
```

Geometric.cpp

```
/*
 * Geometric.cpp
 *
 */

#ifndef GEOMETRIC_H_
#define GEOMETRIC_H_

#include <typeinfo>
#include <iostream>
```

```
#include <string>
#include <ctime>
#include <cmath>
using namespace std;

class Geometric {
        private:
                string color = "White";
                bool filled;
                string dateCreated;
                int id;

        public:
                //delegating constructor
                Geometric() :
                                Geometric(false, "White") {
                }
                Geometric(bool filled, string color) {
                        this->setColor(color);
                        this->setFilled(filled);
                        this->setDateCreated(dateCreated);
                        this->setId();
                        //cout << this->toString() << " created" << endl;
                }
                virtual ~Geometric() {
                //      cout << this->toString() << " destroyed" << endl;
                }
                string getDateCreated() {
                        return this->dateCreated;
                }
                void setDateCreated(string dateCreated) {
                        //create the current date and time
                        time_t now = time(0);
                        //ctime will return a "\n" at the end of the date, using substr from 0 to 24 character
                        this->dateCreated = ((string) ctime(&now)).substr(0, 24);

                }
                bool isFilled() {
                        return filled;
                }
                void setFilled(bool filled) {
                        this->filled = filled;
                }
                string getColor() {
                        return this->color;
                }
                void setColor(string color) {
                        this->color = color;
                }
                int getId() {
                        return this->id;
                }
                void setId() {
                        this->id = (int) rand()%99 +10 ;
                }
                //method to delete the pointer object
                void destroyClass() {
                        //cout << this->toString() << " deleted" << endl;
                        delete this;
                }

                //return general info about Geometric object name, it is virtual so that the subclass can over
                virtual string toString() {
                        return "color: " + this->color + " and filled: " + ((this->isFilled() == 1) ? "true" :
                }
                //return the Geometric class name, it is virtual so that the subclass can override it.
                virtual string getClass() {
                        return "Geometric";
                }

                virtual double getArea() {
                        return 0.0;
                }

                virtual double getPerimeter() {
                        return 0.0;
```

```
            }

};

#endif
```

Triangle.cpp

```
#ifndef TRIANGLE_H_
#define TRIANGLE_H_

#include <iostream>
#include <string>
#include <cmath>
#include <stdio.h>
#include <string>
#include <sstream>
#include <iomanip>
#include <iostream>
#include "Geometric.cpp"

using namespace std;

//--------Analysis and Design----------------------
/*
Analysis
    We have to create a Triangle object that depends on Geometric. This
    Triangle object will override the functions of the Geometric object.
Design
    1. Constructor(s) - We must implement a default (no-arg) constructor
    that sets the side lengths to 1.0, 1.0, 1.0. We must also implement
    a constructor that sets the side lengths to the passed arugments.
    2. In the constructor, we must verify that the provided side lengths
    can create a valid triangle. This means that -
    - no side can be longer than the sum of the other two sides
    - no side length can be negative
    3. Area - in the getArea method we must return the calculated area of
    the Triangle. This can be done by defining variable s as -
    - s = (A + B + C)/2
    Then we can define the area of the triangle as -
    sqrt(s(s - A)(s - B)(s - C)).
    4. Perimeter - We must calculate the perimeter of the triangle by
    finding the sum of the three side lengths. (A + B + C)
    5. toString - we must return the english-based representation of the
    triangle. We have to display the double side length values rounded off
    to one decimal place, as well as calling the superclass toString
    implementation.

*/
//--------Write your code here----------------------
class Triangle: public Geometric {
    private:
        double sideA;
        double sideB;
        double sideC;
    public:
        Triangle(){
            Triangle(1.0, 1.0, 1.0);
        }
        Triangle(double sideA, double sideB, double sideC){
            this->sideA = sideA;
            this->sideB = sideB;
            this->sideC = sideC;

            // valid triangle check
            if((sideA >= sideB + sideC || sideB >= sideA + sideC || sideC >= sideA + sideB) || (sideA < 0 || s
                this->sideA = 1.0;
                this->sideB = 1.0;
                this->sideC = 1.0;
```

```
            }
        }
        double getArea(){
            double s = 0.5 * (sideA + sideB + sideC);
            return sqrt(s*(s - sideA)*(s - sideB)*(s - sideC));
        }
        double getPerimeter(){
            return sideA + sideB + sideC;
        }
        string toString(){
            stringstream pValA;
            pValA << fixed << setprecision(1) << sideA;
            string sideAs = pValA.str();

            stringstream pValB;
            pValB << fixed << setprecision(1) << sideB;
            string sideBs = pValB.str();

            stringstream pValC;
            pValC << fixed << setprecision(1) << sideC;
            string sideCs = pValC.str();

            return "Triangle: side1 = " + sideAs  + " side2 = " + sideBs + " side3 = " + sideCs + " " + Geomet
        }
};
//-------End of your code----------------------------
#endif
```

```
entrypoint.cz
```

```
Driver.cpp
```

Name

Custom test case

Input

1 2 20 blue false

Output (Lines:4)

0.43
3.00
Triangle: side1 = 1.0 side2 = 1.0 side3 = 1.0 color: blue and filled: false

Expected Output (Lines:0)

Status

NA

Name

Custom test case

Input

3 4 5 green false

Output (Lines:4)

6.00
12.00
Triangle: side1 = 3.0 side2 = 4.0 side3 = 5.0 color: green and filled: false

Expected Output (Lines:0)

Status
NA

Name
Custom test case

Input
2 -2 2 red true

Output (Lines:4)
0.43
3.00
Triangle: side1 = 1.0 side2 = 1.0 side3 = 1.0 color: red and filled: true

Expected Output (Lines:0)

Status
NA

Name
Default

Input
2 2 2 red true

Output (Lines:4)
1.73
6.00
Triangle: side1 = 2.0 side2 = 2.0 side3 = 2.0 color: red and filled: true

Expected Output (Lines:4)
1.73
6.00
Triangle: side1 = 2.0 side2 = 2.0 side3 = 2.0 color: red and filled: true

Status
Pass

Name
Invalid Triangle

Input
4 22 5 green false

Output (Lines:4)
0.43
3.00
Triangle: side1 = 1.0 side2 = 1.0 side3 = 1.0 color: green and filled: false

Expected Output (Lines:4)
0.43
3.00
Triangle: side1 = 1.0 side2 = 1.0 side3 = 1.0 color: green and filled: false

Status
Pass
