**C++ Workshop – 150018**

**Homework Assignment #7**

**Polymorphism**

In this question you are asked to define an abstract class **Shape** for representing shape objects. In addition, you will define classes that inherit from **Shape** for representing rectangle, triangle, and circle objects.

1. Define the class **Shape**. The class should include the following data members and methods:
   * **numOfPoints** (int): for the number of vertices of the shape
   * array of **points** (Point\*): for storing the points of the shape (use the Point class as defined in the first tutorial).
   * default constructor
   * constructor which inputs the number of vertices of the shape and creates an array of the appropriate size. In addition, it should ask the user to Enter values of # points: where # is the actual number of points the user needs to enter. It then reads in the appropriate number of points each in the format (x,y) using the operator>>.
   * copy constructor
   * move constructor
   * virtual destructor
   * operator>> for printing out all points in the shape in the format points: (x1,y1) (x2, y2)... (xn,yn)
   * **area**: constant method for computing the area of the shape
   * **isSpecial:** method that returns true if the shape has a special attribute
   * **printSpecial:** method that prints the special attribute
2. Define a class **Circle** which inherits from **Shape** and which represents a circle (center point and radius). The class includes the following:
   * data member radius (float)
   * constructor which receives as input the radius of the circle and builds an array of a single point for the center of the circle (using its base class’s constructor)
   * special attribute: canonical circle when the center is (0,0). The print of such a circle is A canonical circle with a radius # where # is the radius of the circle.

1. Define a class **Triangle** which inherits from **Shape** and which represents a triangle. The class includes the following:
   * default constructor which constructs the vertices of the triangle (using its base class’s constructor)
   * special attribute: equilateral triangle (when the 3 sides are equal). The print of such a triangle is An equilateral triangle with a side length # where # is the length of the sides.

Note the calculation of area of a triangle can be done as follows: let be the angle between sides a and b, then according to the cosine rule,

and the area can be calculated with the sine rule:

1. Define a class **Rectangle** which inherits from **Shape** and which represents a rectangle. The class includes the following:
   * default constructor which constructs the vertices of the rectangle (using its base class’s constructor)
   * special attribute: square (when the 4 sides are equal). The print of such a triangle is Square with a side length # where # is the length of the sides.

Each of the classes you have defined should implement all the relevant methods.

Note some of the methods need to be defined as virtual or as pure virtual.

In addition, duplication of code should be avoided when possible and no code should be rewritten in derived classes that inherit the code. That is, if there is a method or part of a method in the base class, do not write code that performs the same action in the derived class.

Note: The cmath library can be used for mathematical functions.

1. Write a main program that defines an array that contains data for all types of shapes.

How should the array be defined so that the **area**, **isSpecial** and **printSpecial** methods perform what is required for all object types?

The main program should then initialize the array in different ways as follows:

The number of shapes should be read from the user in the following format:

Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4

and create the shapes accordingly (creation of points will be done in the base class constructor).

For a circle, the radius of the circle should be read after printing: Enter radius: If the input is not valid the message: invalid input should be printed.

Afterward, the main program should iterate through all the shapes in a single loop and print the points that define the different shapes (using operator >>) and the area of the shape:  
points: (x1,y1) (x2,y2)…(xn,yn) area is: #

If the shape has a special attribute, the program should then print the description of the attribute.

How many shapes you would like to define?

3

Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4

3

Enter values of 3 points:

(0,0) (1,0) (1,1)

Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4

1

Enter radius:

4

Enter values of 1 points:

(0,0)

Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4

2

invalid input  
Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4

4  
Enter values of 4 points:

(0,0) (1,0) (1,1) (0,1)

points: (0,0) (1,0) (1,1) area is: 0.5

points: (0,0) area is: 50.24

A canonical circle with a radius 4

points: (0,0) (1,0) (1,1) (0,1) area is: 1

Square with side length 1