

CS 3202 – Spring 2014
Assignment 4 - 20 points
Due date: Wed., April 2nd at 9a

Learning objectives

- Demonstrate knowledge of the appropriate use of inheritance, polymorphism, and dynamic binding when writing a program that uses related classes.
 - A major component of this assignment is designing and implementing the class hierarchy. A portion of your grade will be on an appropriate class design.

Specifications

1. Read through the entire project description before beginning.
2. This is an individual assignment do not look at or use another student's code.
3. The project will be a C++ application using Eclipse, so to get started create a new C++ project and name the project *FirstnameLastnameShapeManager*.
4. The application will need to be able to manage the following types of shapes:
 - i. Rectangle
 - ii. Square
 - iii. Equilateral triangle
 - iv. Isosceles triangle
 - v. Scalene triangle
 - vi. Circle
5. The program must allow the storage of a the **color** of a shape as a **string**, its orientation which will be an integer value from 0 to 359, inclusive, and the storage of specific size details as integers for each specific type of shape as follows:
 - a. Circle: diameter
 - b. Rectangle: width, height
 - c. Square: length
 - d. Equilateral triangle: side
 - e. Isosceles triangle: base, sides
 - i. sides will be for the two sides that are of equal length
 - f. Scalene triangle: side1, side2, side3
6. Each class should have the appropriate constructor(s) to correctly initialize all the data required in each class.
7. Any shape class should be able to compute and store its area and perimeter.

8. The application must do the following:
- a. It should query the user for how many total shapes to create. Once the user has entered the number of shapes, the program should randomly determine each shape to be added to the list of shapes, e.g., if the user wants four shapes, it could end up with two squares, a rectangle and a scalene triangle. All the data needed for each shape also must be randomly determined. The data for each shape should be randomly determined as follows:
 - i. The specific color of a shape must be randomly determined from among the following four color strings, "Orange", "Purple", "Red", "Yellow".
 - ii. The orientation of the shape will be determined as a random integer between 0 and 359, inclusive.
 - iii. The size details for each shape should be randomly determined as follows:
 1. Circle: diameter in the range 10-20 inclusive
 2. Rectangle: width, height in the range 5-15 inclusive
 3. Square: length 5-15 inclusive
 4. Triangle: base or any sides in range 20-30 inclusive
 - b. All the shapes should be stored in a collection of data.
 - c. Provide functionality that iterates through the collection of Shapes and prints information out the following information about each shape: the type of shape, the x,y position, color, area, perimeter, and the data that is specific to the shape. For example, for a circle the output would be:

```
Shape: Circle
Color: Red
Orientation: 320
Area: 254.47
Perimeter: 56.55
Diameter: 18
```

Example output for a rectangle would be:

```
Shape: Rectangle
Color: Orange
Orientation: 320
Area: 50.00
Perimeter: 30.00
Width: 5
Height: 10
```

- d. The area and perimeter data need to be printed out to two (2) decimal places.
- e. The functionality to print out all the shapes right after the list of shapes has been created.

Grading rubric

Any program that does not compile will receive a 0. Partial credit is not possible for any program that does not compile.

If a program is only partially complete and a category cannot be accurately assessed you will not receive full credit for that category.

	Exceptional	Acceptable	Amateur	Unsatisfactory
Requirements	8 pts.	6 pts.	4 pts.	0 pts.
Readability/formatting/organization	3 pts.	2 pts.	1 pt.	0 pts.
Reusability	6 pts.	4 pts.	2 pt.	0 pts.
Documentation	3 pts.	2 pts.	1 pt.	0 pts.

Grading description

	Exceptional	Acceptable	Amateur	Unsatisfactory
Requirements	The program works and meets all of the requirements.	The program works and produces most of the correct results and meets nearly all of the requirements correctly.	Program does not meet most of the requirements correctly.	Program does not meet or implement the majority of the requirements.
Readability	The program is exceptionally well organized, very easy to follow, and there are not any compiler warnings.	The code is fairly easy to read and there are a few compiler warnings that were not legitimately explained as to why they still exist.	The code is readable only by someone who knows what the code is supposed to be doing and/or there are many unresolved compiler warnings.	The code is poorly organized and very difficult to read.
Reusability	The code could be reused as a whole or each routine could be reused.	Most of the code could be reused in other programs.	Some parts of the code could be reused in other programs.	The code is not organized for reusability.
Documentation	The required documentation is well written and clearly explains what the code is accomplishing. There is not any redundant inline commenting.	The required documentation is there, but not complete or is only somewhat useful in understanding the code.	The required documentation is missing in several places and is not useful or redundant in nature.	The documentation is poor and very incomplete.

Submission

Export the project from Eclipse and verify its name is *FirstnameLastnameShapeManager.zip* and submit the zip file by the due date.