1 Objective

The purpose of this assignment is to exercise your familiarity with inheritance. For this assignment, you’re required to turn in one program that involves inheritance, classes and arrays. You can use an IDE to write this. However, you’re also required to test it on a terminal before turning it in.

Please email your files TestShapes.java to jayarama@cs.fsu.edu

2 The Program

This program should be called TestShapes.java. For this program you are required to write 5 classes. The requirements are listed below. Please follow the given naming guidelines.

- Define a class called Shape.
  - This class just has one attribute - color.
  - Write a default constructor that sets color to “red”.
  - Write a parametrized constructor and accessor and mutator methods.
  - Write a method called print, that takes no parameters and prints the color.
  - Also define a method called “area” that returns double, but leave it empty.

- Define a class called Square.
  - This class inherits from the Shape class.
  - The class has the attribute sideLength - double.
  - Write a default constructor that sets sideLength to 1.
  - Write the parametrized constructor, accessor and mutator methods. Make sure to invoke the superclass’ constructor appropriately in both of the constructors.
  - Override the print and area methods. In the print method, call the superclass’ print method as well. The print method should then print the sideLength and the area of the square. In the area method, just calculate the area and return it.

- Define a class called Rectangle
  - This class inherits from the Shape class.
  - The class has two attributes length - double and width - double.
  - Write a default constructor that sets length and width to 1.
  - Write the parametrized constructor, accessor and mutator methods. Make sure to invoke the superclass’ constructor appropriately in both of the constructors.
  - Override the print and area methods. In the print method, call the superclass’ print method as well. The print method should then print the length and width, and the area of the rectangle. In the area method, just calculate the area and return it.

- Define a class called Circle
- This class inherits from the Shape class.
- The class has the attribute radius - double.
- Write a default constructor that sets radius to 1.
- Write the parametrized constructor, accessor and mutator methods. Make sure to invoke the superclass' constructor appropriately in both of the constructors.
- Override the print and area methods. In the print method, call the superclass' print method as well. The print method should then print the radius and the area of the circle. In the area method, just calculate the area and return it.

- Define a class called TestShapes.
  - This class should only contain the main method.
  - Accept a number ‘N’ from the user. Then create an array of type Shape of size N.
  - Ask the user to choose between the 3 shapes and enter an integer to denote their choice. If the user enters 1, it is a square. If the user enters 2, it is a rectangle, and if the user enters 3, it is a circle. You may assume that the user will only enter 1, 2 or 3.
  - Read in the required attributes (color, and whatever is needed for the user’s choice. Create an object of the appropriate class according to the user’s choice and attach it to the reference in the array. (You can look at TestInherit.java as a guide).
  - Once the array of objects is created, invoke the print method for each object in the array one by one.

Sample Run

Enter the number of shapes: 3
Enter the choice (Square, Rectangle or Triangle): 1
Enter the color: Green
Enter the side length of the square: 12
Enter the choice (Square, Rectangle or Triangle): 3
Enter the color: Purple
Enter the radius of the circle: 3
Enter the choice (Square, Rectangle or Triangle): 2
Enter the color: Blue
Enter the length of the rectangle: 5
Enter the width of the rectangle: 9

Shape 1:
Color: Green
Side Length: 12
Area: 144

Shape 2:
Color: Purple
Radius: 3
Area: 28.27

Shape 3:
Color: Blue
Length: 5
Width: 9
Area: 45

Generic Guidelines

- Please make sure you’re only using the concepts already discussed in class. That is, please try and restrict yourself to loops, selection statements methods, arrays, and classes and objects.

- Please make sure that you’re conforming to specifications (program name, print statements, expected inputs and outputs etc.).
• Please make sure your code is readable.

• Please make sure you’ve compiled and run your program before you turn it in. Compilation errors can be quite costly.