Homework A3: Paint

COSC150

**Due Tuesday Oct 17th, 2017, 11:59PM**

# Description:

The goal of this homework is to practice object-oriented programming principles for a graphical user interface program.

You’ll be writing a program, called Paint. Your program should allow the user to draw a few types of shapes, including rectangle, oval, line, and triangle at arbitrary points on the screen. The user should be able to select the color to draw any of these shapes.

The user should also be able to adjust the shapes by dragging the shapes.

The user should also be able to click buttons to ask for the sum of the areas and the sum of the perimeters over all of the shapes, except for lines, displayed on the screen.

Your program should have a place to display the shapes. You need to provide buttons for the user to select the shape type. For instance, if the user selects a shape type to be rectangle or oval or line, the user should be able to draw rectangles via dragging the shape. If the user selects a shape type of triangle, the user then makes 3 clicks for the corners (no dragging for this one). The user should also be able to select a color from a color palette BEFORE each shape is created. Your program should also have two additional buttons once they are clicked to show the total area and total perimeter of all the shaped displayed in the screen, respectively.

To summarize, you should:

* Design before you code. You are expected to work out the use case diagram, class diagram and sequence diagrams before you code.
* Write Rectangle, Oval, Triangle, and Line classes, all of which will extend a class called Shape.
* The main() should be in the Paint class.
* Your program should display all the shapes that the user wants to draw.
* Your program should be able to provide buttons for the user to select the shape type.
* Your program should provide a color palette for the user to select the shape color.
* Once a shape is selected, your program should allow the user to adjust its dimensions by dragging the shape (to make it bigger or smaller or distorted).
* Your program should provide buttons for the user to tell the total area and total perimeter.

## Other Requirements

* This is a group assignment. Your grades will be shared among all of your team members. A separate peer review survey will be done at the end of the semester. Between different groups, you can explain concepts to each other, but do not hammer out the code together.
* Your program should not crash. It may exit gracefully (e.g., with a polite error message) if it encounters an error -- e.g., an undefined shape type.

**What to turn in:**

* UML use case diagram
* UML class diagram
* UML sequence diagrams (one for each use case)
* You will also need to turn in your codes. Please turn in a single file archive (kinda like a .zip file) that contains your entire program.  It should include everything necessary to compile on its own.  To generate this file, do the following:

1. Remember to put down your name and your netid in each file that you submit.

1. Right-click package name, select “Export…”
2. Select “Java -> Jar File”
3. In the Jar File Specification wizard:
   * 1. **MAKE SURE THAT YOU SELECT “Export Java source files and resources”.**  We will not award any credit without the source Java files.
     2. Save the file as “hw3.jar”.  You may want to use “Browse…” to easily figure out where Eclipse is saving the file.
4. Then, submit hw3.jar to Blackboard.

**Where to go for help:**

For questions about this assignment, please post to Piazza at <https://piazza.com/georgetown/fall2017/cosc150/home>

# **A Grading Rubric**

|  |  |
| --- | --- |
| **Amazing Feat** | **Points Awarded** |
| You submitted something.  Anything. | 10 |
| It compiles! | 20 |
| Shapes are correctly drawn (4 shapes) | 40 (10 for each shape) |
| Shapes are correctly adjusted (4 shapes) | 40 (10 for each shape) |
| Color palette is working correctly | 10 |
| Shape selection buttons are working correctly | 10 |
| Total area correctly computed and outputted | 10 |
| Total perimeter correctly computed and outputted | 10 |
| Correctly used inheritance | 5 |
| Correctly used public, protected and private interfaces | 10 |
| Obeyed The Law of Demeter | 5 |
| Correct Use Case Diagram | 5 |
| Correct Class Diagram | 5 |
| Correct Sequence Diagrams | 10 |
| Correctly used Java Swing classes | 10 |
| **Total** | **200** |

Please note that we may deduct points for clarity, managing dependencies, good naming conventions, and other violations of good design principles and programming practices (as covered during lectures).

# 