

# Assignment 10

## CSc 210 Fall 2017

### Programs Due November 9th 13th, 8:00 pm MST

#### Introduction

In class we have been looking at GUI's and how to create them using JavaFX. For this assignment, you will build upon your shape drawing program. Instead of drawing shapes on stdout, you will draw them onto the screen.

You will turn in your code via github classroom. Here is the link to receive your repository: <https://classroom.github.com/a/Z9Dc0HWY>

For each Java program assigned, you will turn in a .java file for that program. We should be able to run "javac *file.java*" and then "java *file*" on lectura and your program should compile and run successfully.

Your Java programs should adhere to our style guidelines we will give to you. Here is the link: <https://www2.cs.arizona.edu/classes/cs210/fall17/StyleGuidelines.pdf>

#### Drawing Shapes

As stated before, this assignment will draw various shapes on the screen. Your program will create a canvas to draw on that is 800x800 pixels in size. Then, you will use the same REPL from the last assignment to read in commands and draw shapes.

In your REPL, you will handle all the commands defined on the last assignment (some commands have slight changes) and a few new commands. Additionally, you may assume all commands given to your REPL will be on their own line and invalid commands will be on a line by themselves (with no additional arguments). You can find all the new information you need below.

- Your define commands will no longer take in a fill character. Instead, in the same argument position, they will now take in a color. This color could be the string BLACK, BLUE, BROWN, GOLD, GREEN, ORANGE, PINK, PURPLE, RED, or YELLOW which each correlate to filling the shape with that color as defined by the Color class. You may implement more colors if you like, but your code must implement at least these.
- You will implement a CLEAR command. This command takes no arguments. All it does is it erases all shapes from the screen.

- You will also implement a QUIT command which will cause the program to exit with a status of 0.
- The triangle will have a base **b** size that depends on its height **h**. Use the formula  $b = 2h - 1$  which will give a size similar to the text version of the triangle from the last assignment.
- The hourglass is two triangles with heights calculated the same as in assignment 9.
- The top of the parallelogram is shifted to the right by the height. This also simulates the shape of the parallelogram from assignment 9.
- The DUMP now prints out the color value in hex in place of the fill char. You can get this by calling a color object's toString() method.
- For DRAW, you no longer print out a warning message if a shape goes out of bounds while drawing.

To get your REPL and GUI to work at the same time together, you will need to start the REPL in its own thread of processing. We will discuss how to do this in class and section. Additionally, you can [look at this code example](#) which creates a GUI and reads input in from a separate thread for reference.

There is example input and output provided in your repo. There is also a screenshot which shows you what the shapes in that example input should look like.

You are required to turn in at least the following files: Drawing.java (This shall contain your main method), ShapeREPL.java (contains the code for your REPL), Shape.java, Rectangle.java, Triangle.java, Parallelogram.java, and Hourglass.java.

## Miscellaneous

**In order to receive a grade**, your repo must only contain your source java files in the root directory of your repo.

This assignment will be submitted through github classroom. Make sure all of your code you would like to submit is in your repository when the due date arrives.

Note: Your output must match what is defined here in the spec. We will give you a small selection of test cases so you can make sure you have the right format. Do not print extraneous output, you may lose a lot of points (This includes prompts when reading in input!). **These test cases can be found in your repo.**

Remember, do not cheat! Refer to the syllabus and first lecture for more information.