Programming Project 4
60 points

Submission Instructions
Open Eclipse and create a Java Project called *Project4*. In this project create a package named *turtle*. Add a class named *TurtleDriver* and a class named *Turtle* to this package.

At the top of each file, enter a comment with your name, the assignment number, and the date.

When you are finished, export your project from Eclipse and upload it to Canvas before the due date. To do this, right click on the project name and select Export. Select General->Archive File and click Next. Select the project you wish to export and click Browse to browse to a location to save your file. Name this file *YourLastNameYourFirstNameProject4*. Upload this file to Canvas.
Remember that late assignments are not accepted in this course.

Assignment
In this project, we will practice creating and using a class by creating a simplified version of turtle graphics (https://en.wikipedia.org/wiki/Turtle_graphics). We will allow the user to type commands that will control a simple turtle (or pen) and produce a drawing on the canvas.

Your project will have two classes: TurtleDriver and Turtle.

The Turtle class should contain the following private instance variables:

- xCoord, a double that stores the turtle's current x-coordinate
- yCoord, a double that stores the turtle's current y-coordinate
- direction, a double that stores the turtle's current direction (angle) in degrees
- penDown, a boolean that stores whether the pen is down or up
- color, a variable of type Color that stores the current color of the pen (**Hint:** you will need to import java.awt.Color.)

The Turtle class should contain one constructor. This constructor should be a no-argument constructor. This constructor should initialize the canvas size and coordinate system. It should then set the turtle's position to the center of the canvas and the turtle's direction to 0 degrees. The color instance variable should be initialized to black and the pen should be up. This constructor should also draw the turtle's initial position.

The Turtle class should contain the following public methods:

forward (int steps) :

The turtle should move the indicated number of steps in its current direction. The turtle should leave a footprint (a circle) for each step. If the pen is down, a line should also be drawn.

Hints:
1. This is the only method in which you should draw!

2. Here is a quick refresher on the trigonometry you will need for this method:

The x- and y-coordinates of the points on a circle can be described by the following equations:

x = centerX + radius * cosine(angle)

y = centerY + radius * sine(angle)

3. The Java Math class has three static methods that you might find useful: Math.sin(double angleInRadians), Math.cos(double angleInRadians), and Math.toRadians(double angleInDegrees). Refer to the Math class's Javadoc for more information.

right (double angle):

The turtle should turn the indicated number of degrees to the right.

left (double angle)

The turtle should turn the indicated number of degrees to the left.

penUp()

The turtle should NOT draw a line when it changes location.

penDown()

The turtle should draw a line when it changes location

setPenColor(String color)

The pen color should be changed to the indicated color: black, red, green, yellow, or blue.

Your driver program (i.e. the main method of the TurtleDriver class) should prompt the user to enter one of the following commands:

forward steps (for example, forward 5)
right angle (for example, right 45)
left angle
penup
pendown
pencolor color (for example, pencolor blue)
quit

Your program should stop accepting commands when the user enters the quit command. To terminate the program you will need to close the StdDraw window.

Sample Output
Here is a link to a video that shows the program in action:

https://youtu.be/6XXuM14d2Wo

Grading Criteria (60 points possible)

Points	Criteria
0-5 points	**Input:** Does the driver program prompt the user for a command until the user enters quit? Does the driver program correctly read input from the user? Does the driver program indicate when an invalid command has been entered?
0-10 points	**Correctness:** Does the driver program correctly call methods in the Turtle class based on commands entered by the user?
0-2 points	**Correctness:** Does the Turtle class include all private instance variable described above?
0-3 points	**Correctness:** Does the Turtle's no argument constructor properly initialize all instance variables? Does it set the canvas's size and scale? Does it display the turtle's initial position?
0-10 points	**Correctness:** Does the Turtle class correctly implement the *forward* method?
0-5 points	**Correctness:** Does the Turtle class correctly implement the *right* and *left* methods?
0-5 points	**Correctness:** Does the Turtle class correctly implement the *penUp* and *penDown* methods?
0-5 points	**Correctness:** Does the Turtle class correctly implement the *setPenColor* method*?*
0-10points	**Output:** Are the turtle's movements properly displayed on the Canvas? Are they displayed in the correct color?
0-5 points	**Style:** Is the code easy to read? Is the code indented in a style similar to that shown in the textbook? Are blank lines used to divide the code into sections? Is a comment with the required information included at the top of the file? Are comments used to provide details that are not obvious? Are meaningful variable names used? (See Section 1.4 of the textbook for documentation and style guidelines.)