**Lab 99 – Graphics**

Open BlueJ, and create a new BlueJ project titled **Lab99-Graphics** in your CS\LABS folder.

Create a new class with a main method called **PracticeProblems**:

In this lab, we will learn how to make drawings and animations in Java. We will also learn how to manipulate objects on the screen using keyboard or mouse inputs. Java graphics can be VERY complicated, so we are going to utilize a class called StdDraw (Standard Draw). StdDraw is not a part of the Java that is already installed on your computer. It’s a class coded by Robert Sedgewick and Kevin Wayne of Princeton University.

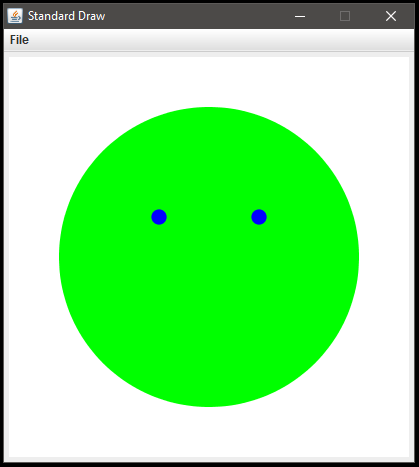
**Getting StdDraw (Standard Draw) into your project:**

* You need a copy of StdDraw.java in your Lab99-Graphics project. There are multiple ways to do this.
  + Download the [StdDraw.java](https://drive.google.com/open?id=1_Q_3qGMp8VGMS4lZvbtVenksioMqGFlW) file to your computer and drag it into your Lab99-Graphics project.
  + OR, create a new class in your project called StdDraw. Click this link to [StdDraw.java](https://drive.google.com/open?id=1_Q_3qGMp8VGMS4lZvbtVenksioMqGFlW) and copy/paste all the code from the link into your StdDraw class in your project.

**Before each problem, insert a COMMENT with the problem number.**

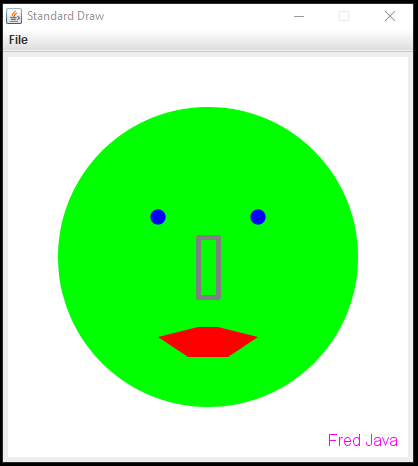
1. Set your drawing canvas to a size of 400 pixels by 400 pixels. Set your Xscale to a range of 0 to 400 and your Yscale to a range of 0 to 400.
2. We are going to use the various shapes we learned to make a crude smiley face. Use the filledCircle method to draw a face (in whatever color you’d like) centered at (200, 200) with a radius of 150.
3. Use the point method (not the circle method) to draw two points for eyes in the color of your choice. You will need to use a guess & check strategy to figure out the PenRadius and the locations of the points. It doesn’t have to be perfect.

With the circle and 2 points, your drawing should now look something like:

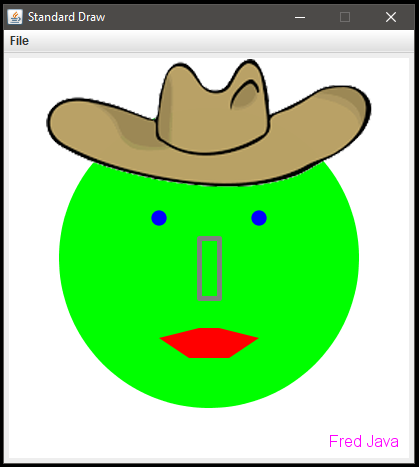


1. Use the rectangle method to draw an (unfilled) nose in the color of your choice.
2. Use the filledPolygon method to draw RED lips. Your polygon should have at least 4 points. You will need to first build your xValue and yValue arrays. This will take quite a few adjustments to get right.
3. Add text to display your character’s name in the bottom right corner in the color of your choosing.

Your drawing should now look something like this:



1. Let’s give our person a cowboy hat by loading an image file into our drawing. Download the file [cowboyhat.png](https://drive.google.com/open?id=11yW-aqsyjPMimWW6Y0gyz3OIzb_RIfT7) from Google Drive into your BlueJ project folder. (It has to be in the same location directory as your PracticeProblems.java and StdDraw.java.)

****

**My Drawing**

Lots of practice in placing elements within a drawing canvas will be very helpful to you when you create more complicated animations and games later on. Being able to visualize the x, y coordinate system in your mind is very valuable, so let’s get some extra practice.

Create a new class called MyDrawing.

Set up your own canvas to whatever size/color you’d like. Create your own masterpiece! Utilize shapes, text, and/or image files to get it exactly how you want it.

**My Animation**

Now that you’ve mastered drawing, creating animations shouldn’t be too tricky. Just follow this basic pseudocode blueprint.

* Set up your drawing canvas
* Turn on double buffering\*\*
* Start a loop. Each iteration is responsible for drawing a frame of your animation.
  + Display your drawing
  + Wait a few milliseconds
  + Make minor changes to your drawing so next iteration it will be drawn differently

\*\* double buffering allows all your drawing commands to take place offscreen. Once the drawing is ready to be viewed, you just use the show() method to display it all at once instead of piece-by-piece. This makes your animations smoother.

Download the [AnimationDemo.java](https://drive.google.com/open?id=1teRo8w4YI_3n9uh4DvteYYqJnlz6hjOQ) file to your computer and drag it into your Lab99-Graphics project to see an example animation. Play with the variables to see what they all do.

Create a new class with main called **MyAnimation** and see what neat animations you can make on your own.