Lab 3: Looping Turtles

In lecture this week, we created some turtle art: Spirograph like patterns and stars with Python’s “turtle” class. I have posted the python source code for that star program on the class website. The example below also uses this technique to create a simple spiral pattern.

```
1) import turtle
2)
3) turtle.pensize(2)
4) turtle.pendown()
5)
6) length = 0
7) for i in range(30):
8)   length = length + 5
9)   turtle.forward(length)
10)  turtle.right(91)
11)
12)  turtle.exitonclick()
```

Your task is to make more complex and interesting patterns that are creative also beautiful to behold.

In the example above, a variable was used in the turtle.forward command and that variable’s value is changed in each iteration of the loop. The same can be done for the red, green and/or blue values in the color statement:

```
turtle.color(r, g, b)
```

to produce color effects that gradually change from dark to light or shift hue.

Patterns with straight lines and sharp turns are easy to make to Turtle Graphics. If you happen to know trigonometry and want to make the loops and curves of actual Spirograph (that is, the patterns made by movement of a small circle that rolls, without slipping, on the inside of a rigid circle) then check out: “Plotting the Spirograph equations with gnuplot” by Víctor Luaña. This website gives instructions for gnuplot, but the equations developed are language independent.
Grading Rubric [20 points total]:

[A: 2 points]: Attach one file in blackboard learn:

```
CS105_Lab3_yourFirstName_yourLastName.py
```

The first line of your program must be a comment that includes your name.

[C: 2 points]: Your program must include comments that delineate which lines of code are used to draw which elements of the image.

[D 10 points]: When run in python 3.4 using PyCharm, your program draws a spectacular design using at least 1000 draw commands (commands that actually move the turtle with the pen down). For example, `turtle.forward()` and `turtle.goto(x,y)` are draw commands. By contrast, `turtle.right()`, while important, is not create a line on the screen.

In the stars on a stick example we created in class, the program creates a list of many turtles. You may do that - or not. One turtle can draw 1000s of lines into an interesting image without help (other than accurate instructions). Other patterns are best made with turtles in synchronized cooperation - your choice.

[E 4 points]: Your program draws in at least 100 different colors.

[G 2 points]: Your program does not exit when finished drawing, but waits for the user to click someplace in the window. Then it exits cleanly. when run in. This will be tested in PyCharm using Python 3.4. To do this, make sure the last line that executes in your program is:

```
turtle.exitonclick().
```

Extra Credit 10 Points: Sheer Awesomeness!!!

To get this extra credit, you need to do three things:

1) Your program produces an exceptionally interesting pattern.

2) Send e-mail to me (joel@ unm. edu) with the subject: “CS-105 Lab 3 Sheer Awesomeness!!!” In the body of the e-mail, just place your name - which I will use to find your submission in Blackboard Learn.

3) If you get an e-mail back from me saying ”yes, you are awesome”, then I will have your program ready on my laptop to show to the class. When I call out your name, come up and give a quick demo of your awesomeness.