

CS1110 Lab 3 (Feb 10-11, 2015)

First Name: _____ Last Name: _____ NetID: _____

The lab assignments are very important and you must have a CS 1110 course consultant “tell CMS” that you did the work. (Correctness does not matter.) This can be done any time up until the start of the next lab (Feb 24-25). Thus, if you have trouble with a problem, then you have a 2 weeks to get help from the teaching staff. But it is in your interest to complete this Lab before the A2 due date (Feb 13). If you finish before the hour is over, then you can leave early or you can work on the current assignment. Indeed, you are not required to physically attend the labs at all. Just make sure your work is “checked off” by a consultant. And remember this: *The lab problems feed into the assignments and the assignments define what the exams are all about.*

1 Getting Set Up

From the Lab webpage, download `DemoGraphics.py`, `Turkey.py`, and `DemoMath.py`. From the Lecture webpage download `simpleGraphics.py`. Put them all in the same folder, say `Lab3`. In the command shell, navigate the file system so that this folder is THE CURRENT WORKING DIRECTORY.

2 DemoMath

This part of lab is about the module `DemoMath.py`:

```
# DemoMath.py
# Examines a function that computes approximate square roots
def sqrt(x):
    """Returns an approximate square root of x as float.

    Performs five steps of rectangle averaging.

    Precondition: The value of x is a positive number."""
    x = float(x)
    L = x
    L = (L + x/L)/2
    L = (L + x/L)/2
    L = (L + x/L)/2
    L = (L + x/L)/2
    L = (L + x/L)/2
    return L

# Application Script
if __name__ == '__main__':
    """ A keyboard input framework for checking out sqrt.
    """
    import math
    x = input('Enter a number whosesquare root you want: ')
    y1 = math.sqrt(x)
    y2 = sqrt(x)
    print '\n\n      x = %5.2f' % x
    print 'math.sqrt(x) = %15.12f' % y1
    print '      sqrt(x) = %15.12f' % y2
```

1. What is displayed if you ask for the square root of 9? What is displayed if you ask for the square root of 900?

2. Go into Komodo and double the number of `L = (L + x)/2` statements. Save the updated `DemoMath.py`. Now what is displayed if you ask for the square root of 9? What is displayed if you ask for the square root of 900?

3. Does it make a difference if the `import math` statement moved to a position just before the definition of `sqrt`?

4. What happens if you say `y2 = DemoMath.sqrt(x)` instead of `y2 = sqrt(x)` in the Application Script? Explain.

5. What happens if you remove the `return` statement in `sqrt`? Explain.

6. Why is it necessary to have the statement `x = float(x)` in the function body of `sqrt`?

7. How could `sqrt` be modified so that it could handle the input `x = 0`?

8. Unindent something and explain what Python says.

9. What happens if you remove the funny `if __name__ == '__main__':` line?

3 Designs

Here is a quick synopsis of the three graphics procedures in `simpleGraphics`:

```
DrawRect(x,y,L,W,color=c,stroke=s,rotate=d)
DrawDisk(x,y,r,color=c,stroke=s)
DrawStar(x,y,r,color=c,stroke=s,rotate=d)
```

In each case `x` and `y` define the center. Disks and stars have a radius (that's `r`). For fill color, `c` must be a built-in color like `YELLOW` or an rgb array like `[.2,.3,.4]`. If no color is specified, then there is no fill and the object is “transparent.” The parameter `stroke` controls how much the edge is highlighted. Typical values for `s` are 0, 1 (the default), 5, 10. To rotate a star or rectangle counterclockwise set `d` to be the degree measure of the rotation (default is 0.)

Lecture 5 has lots of “how to” examples that illustrate the five functions and built-in colors that are part of `simpleGraphics`. But you can also get details via the Python help facility. To illustrate, get in the Python interactive mode and enter

```
>>> from simpleGraphics import *
```

Now you can use the “help” facility like this

```
>>> help(DrawRect)
```

to learn about the various functions in `simpleGraphics`.

Using the help facility, what does `MakeWindow(10,labels=False,color=PURPLE)` do in English?

3.1 DemoGraphics

The module `DemoGraphics.py` looks like this:

```
# DemoGraphics.py
""" Draws a design with squares and a design
with rings."""

from simpleGraphics import *

# First Figure
MakeWindow(6,bgcolor=DARKGRAY,labels=False)
DrawRect(0,0,6,6,color=CYAN,stroke=5,rotate=0)
# Add more squares...

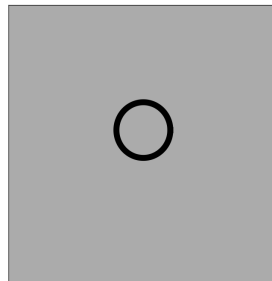
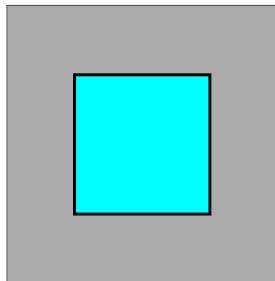
# Second Figure
MakeWindow(10,bgcolor=DARKGRAY,labels=False)
# Rings
DrawDisk(0,1,2,stroke=10)
# Add more rings...

ShowWindow()
```

From the command shell, run `DemoGraphics.py`:

```
> python DemoGraphics.py
```

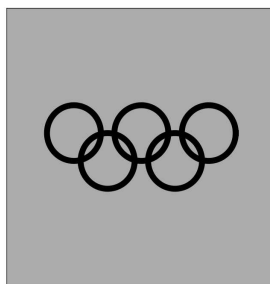
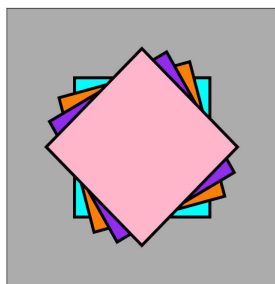
Observe that it displays a square in one figure and a ring in a second figure-



If you want to run this script again *you must close both figure windows*. To close a figure window, click on the little “x box” that you see in its upper right corner. Thus, the standard work pattern when you are developing graphics python code is

- Step 1. Edit the module in Komodo.
- Step 2. Save the module.
- Step 3. Run the module from the command shell.
- Step 4. Look at the displayed figures.
- Step 5. Close the figures and go back to Step 1.

With that in mind, modify `DemoGraphics.py` so that it produces these two figures:

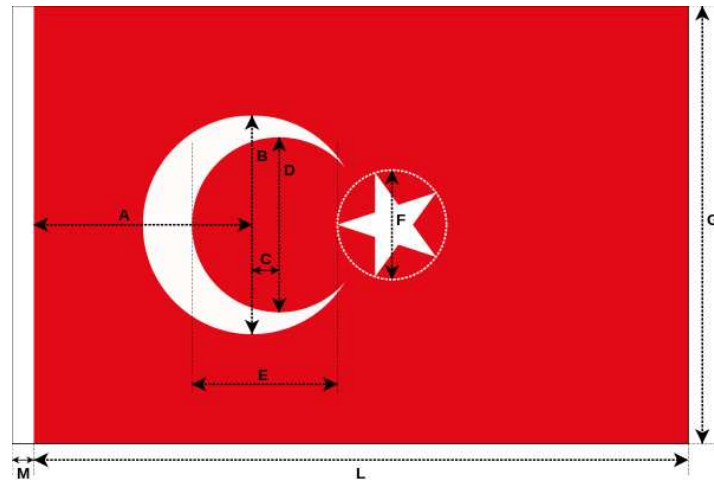


For the squares, you will need three more calls to `DrawRect`. These should produce squares with the same size, the same center, and the same perimeter highlighting. But they are to be colored differently and rotated. The overall design has four squares. The second square should be `ORANGE` with rotation 15 degrees. The third square drawn should be `PURPLE` with rotation 30 degrees. And the fourth square drawn should be `PINK` with rotation 45 degrees. How many squares would you see if the `PINK` square had a rotation of 90 degrees? Why?

For the Olympic rings you must add four more rings. (Note that a “ring” is just a disk with no fill color and a thick perimeter.) Obtain the top row of rings by adding in one ring to the left of the given ring (shift left by 5 units) and one ring to the right of the given ring (shift right by 5 units). After that is working, add in the bottom two rings. Their centers are on the horizontal line $y = -1$ and their centers are 5 units apart from each other.

3.2 Turkey

Here are the specs for the flag of Turkey courtesy of Wikipedia:



| Letter | Measure | Length |
|--------|--|--------|
| G | Width | G |
| A | Distance between the centre of the outer crescent and the seam of the white band | $G/2$ |
| B | Diameter of the outer circle of the crescent | $G/2$ |
| C | Distance between the centres of the inner and outer circles of the crescent | $G/16$ |
| D | Diameter of the inner circle of the crescent | $2G/5$ |
| E | Distance between the inner circle of the crescent and the circle around the star | $G/3$ |
| F | diameter of the circle around the star | $G/4$ |
| L | Length | $3G/2$ |
| M | Width of the white hem at the hoist | $L/30$ |

Take a look at the module `Turkey.py`. It houses a function for drawing the Turkish flag an an application script that can be used to check it out. By entering

```
>>> python Turkey
```

we get this:



Obviously, `DrawTurkey` is not properly locating the white rectangle, the white disk, the red disk and the star. And they are all improperly sized. And the star is not tilted. Your job is to correct these flaws using the specs given above as your guide.