CSI 11 **In-class activity** **Day 17**

Turtle, loops and decision structures

1. Let's write a program when the user is asked to enter a positive integer.
If the entered integer is even, let's draw a circle in the middle of the graphics window, and if the integer is odd, let's draw a square.

Here is a draft of the program: follow the directions in the comments

```
from turtle import *

window = Screen() # open a graphics window

window.screensize(800,600) # re-set the graphics window size to
# 800 pixels wide and 600 pixels high

lina = Turtle() # create a drawing turtle named lina

number = ??? #get a positive integer from the user, do not forget
# to convert the input value to an integer, use int()

if number > 0: # we make sure that the user entered a positive value
  if ??? # check that the the number is even
    ??? # draw a circle

  else: # this is the case when the number is odd
    ??? # draw a square

else:
  # this is the case when the entered value is not positive,
  # display a message in this case (in Python shell)

window.exitonclick() # when the user clicks anywhere
# inside the graphics window it will be closed
```

Show the program to the instructor or in-class tutors to get it marked as complete.

2. Consider the program **TurtleWithManyRepetitions.py**
You can see that a lot of code is repeated there. Improve the program by replacing repeated code with a loop.

Show the program to the instructor or in-class tutors to get it marked as complete.

3. Consider the program that tries to draw a pyramid: **Turtle_pyramidInTheWorks.py**.
So far it only has two bottom levels. I want it to have 11 levels.
Use loops to get 11 levels.

Show the program to the instructor or in-class tutors to get it marked as complete.

Homework assignment:

1) Turtle race: http://interactivepython.org/runestone/static/thinkcspy/Labs/lab03_01.html

2) Grab the program **Turtle_frame.py** from our web site
I want to draw a frame in the graphics window and I already started writing the program.
Finish drawing the frame.

3) Let's create a spiral pattern. You have several options:

(1) start at the center of the screen (the origin)

(2) start at one of the corners (say lower left) of the screen.

In both options, use `while loop`.

Idea: start with some length, say 300 pixels, `length = 300`
Run a while loop. At each iteration of the while loop: draw the line of `length`, then decrease the `length`, and turn 90 degrees.

Send all three programs to my e-mail: **natna20@gmail.com**