CS100 Fall 2015

Name _____

CPADS Assignment 3 – Due 9/21

"Up and Down!"

The goal of this section of the course is to introduce fundamental programming constructs using a simple scripting language, Python. This approach will allow us to focus on *programming* rather than *syntax*, i.e. formulating a procedural solution. To accomplish this task we may write both *console* programs that process text files, as well as *turtle graphics* programs where we draw graphics in an "Etch-a-Sketch" fashion.

1. Slow and Steady

For this assignment we will use a turtle graphics library known as *Swampy* (http://www.greenteapress.com/thinkpython/swampy/). In the turtle graphics world, we move a virtual turtle around the screen using only a few simple commands (hence *planning* will be important). Additionally, the turtle can pick *up* or put *down* the pen. The commands are:

`fd(t, length)` – moves turtle `t` forward `length` units
`bk(t, length)` – moves turtle `t` backward `length` units
`lt(t, angle)` – turns turtle `t angle` degrees to the left
`rt(t, angle)` – turns turtle `t angle` degrees to the right
`pd(t)` – starts drawing for turtle `t` (pen down)
`pu(t)` – stops drawing for turtle `t` (pen up)
`world = TurtleWorld()` – creates a new turtle graphics window
`t = Turtle()` – creates a new turtle in the center of the window pointing right

Name _____

In the program below, **`from TurtleWorld import *`** tells Python to import the entire TurtleWorld library which is needed for this program. The line **`world = TurtleWorld()`** is used to create a turtle graphics window. The line **`crush = Turtle()`** creates a new turtle and assigns it to the variable **`crush`**. The next group of lines then issue movement commands to **`crush`** to perform the drawing. Finally, the last two lines simply keeps the turtle graphics window open until we press enter (in the panel at the bottom of PyCharm) to close it. Lines that begin with the # character are comments and can be used to document your program. Commented lines are not executed and can contain anything.

```
# Load TurtleWorld functions
from TurtleWorld import *
from math import *

# Function to draw a square CCW
def square(t,x):
       pd(t)
       fd(t,x)
       lt(t,90)
       fd(t,x)
       lt(t,90)
       fd(t,x)
       lt(t,90)
       fd(t,x)
       lt(t,90)
       pu(t)

def main():
       # Create TurtleWorld object
       world = TurtleWorld()
       # Create Turtle object
       crush = Turtle()
       crush.delay = 0.001

       # Draw graphics
       square(crush,10)
       lt(crush,45)
       fd(crush,sqrt(2)*10)
       rt(crush,45)
       square(crush,20)
       lt(crush,45)
       fd(crush,sqrt(2)*20)
       rt(crush,45)
       square(crush,30)
       lt(crush,45)
       fd(crush,sqrt(2)*30)
       rt(crush,45)

       # Press enter to exit
       key = input('Press enter to exit')
       world.destroy()

main()
```

- Beside the code above, sketch what the output will be

Name _____

2. Check it out

Now that you have determined what the program in part 1 does, we will check by simply *running* the program.

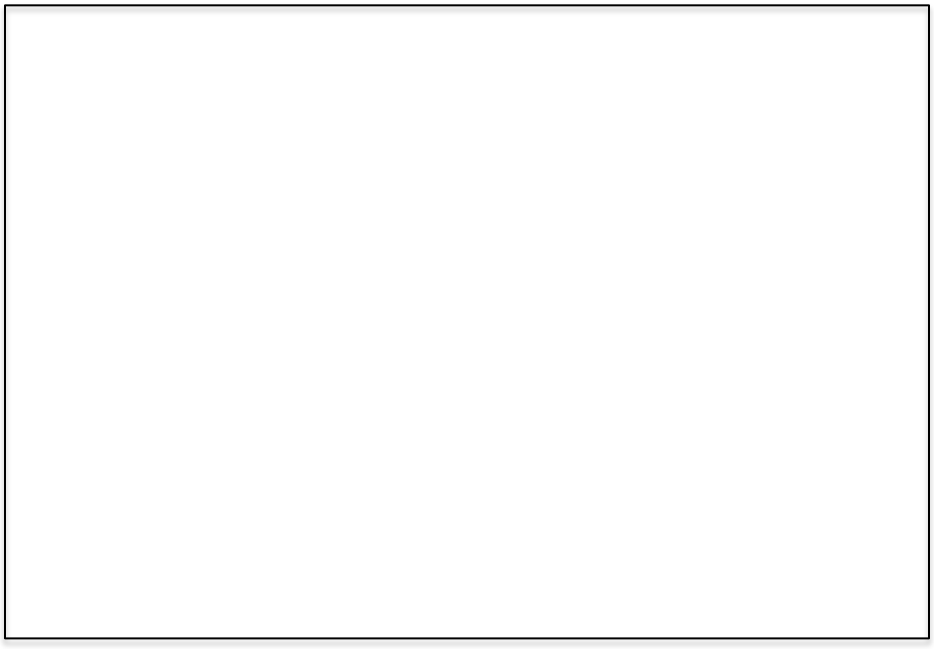
- Open PyCharm (**Menu->Programming->PyCharm**).

- In PyCharm, create a new project by selecting **File->New Project…**
  When the **Create New Dialog** windows pops up:
  - name your project **Assign3**
  - set the location for **Assign3** to be the same **CS100** directory you created during the last activity
  - ensure that the Interpreter is set to Python 3.4.0
  - click **ok** to create the new project

- Download the Python file **`stairs.py`** from the course webpage and save it to the **Assign3** directory
  - Select **`File->Open`**… navigate to the **Assign3** directory and select the **`stairs.py`** file
  - This should open the file in the editor panel (you may need to expand the project tab in the left panel to show the file)

- Run the program by selecting **`Run->Run`**… and then selecting **`stairs`** in the pop-up box that appears.

  Sketch the output produced in the TurtleWorld graphics window in the space below. When done, click in the console window panel at the bottom of PyCharm and press **Enter** to close the TurtleWorld window.



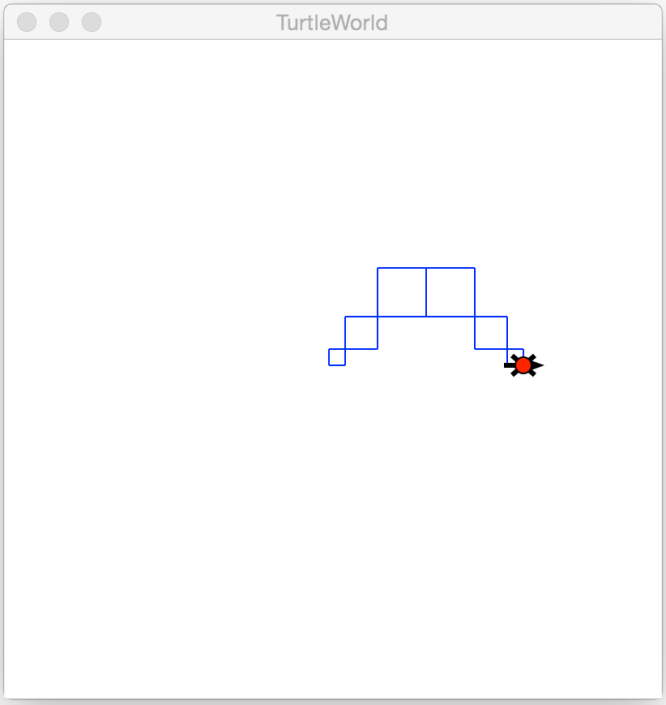
Name _____

3. Get back down

The program you downloaded has several *magic numbers* that will make the program both difficult to read and maintain. In this part, you'll make a few changes to the previous program to remove the *magic numbers* and do some additional drawing.

- Declare a variable **x** below `#Draw graphics` and initialize it using an assignment statement to the value **0**
- Remove the magic numbers in the code for the size of the square (including the movement along the diagonal) by replacing the values with **x**
- Insert the necessary assignment statements that increment **x** by **10** such that the output is identical to the original code
- Show your instructor the new code that uses **x** and appropriate increment operations demonstrating the proper output

- Modify **stairs.py** to draw similar squares mirroring the original three as shown in the figure below.
    - Your code must use **x** for the size of squares and movements between squares.
    - Consider reorienting the cursor before drawing the first new square. This will allow you to use a similar repetitive sequence of instructions by *decrementing* **x**.



CS100 Fall 2015

Name _____

- Submit your source file through Marmoset.
  - Open a web browser (**USE** Google Chrome) and enter the following URL (continue to the website if it brings up a certificate error page)

    https://cs.ycp.edu/marmoset/

  - Enter your login information which you should have received in an e-mail (you probably should change your password to match your YCP account)
  - Select **CS100: Computer Science Practice and Design Studio**
  - Select the **submit** link under **web submission** for **assign3**
  - Click **Choose File…** , navigate to your program directory and select your **stairs.py** file (do not worry about the instructions for jar and zip files).
  - Click **Submit project!**