CPADS Programming Activity II – Due 9/18

“Super Turtle!”

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.**

We will now write our first substantial Python program. For this program we will use a turtle graphics library known as *Swampy* (<http://www.greenteapress.com/thinkpython/swampy/>). In the turtle graphics world, we move a “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)

* Open IDLE (**Start->All Programs->Python 2.7->IDLE (Python GUI)**).
* Open a new editor window (**File->New Window)** and type the following code:

**# Load TurtleWorld functions**

**from TurtleWorld import \***

**def main():**

**# Create TurtleWorld object**

**world = TurtleWorld()**

**# Create Turtle object**

**turtle = Turtle()**

**# Draw graphics**

**fd(turtle,100)**

**rt(turtle,90)**

**fd(turtle,100)**

**rt(turtle,90)**

**# Press enter to exit**

**key = input(‘Press enter to exit’)**

**world.destroy()**

**main()**

* Save the program somewhere in your documents folder (e.g. Documents/CS100/programs) with the filename **rightang.py**
* Run the program by **Run->Run Module** (or using <F5>)

Sketch the output produced in the turtle graphics window.

To understand a bit more about this program, **from TurtleWorld import \*** tells Python to import the entire TurtleWorld library which is needed for this program. The next line **world = TurtleWorld()** is used to create a turtle graphics window. The following line **turtle = Turtle()** creates a new turtle and assigns it to the variable **turtle** (**NOTE: be careful of the capitalization!)**. The next four lines then issue movement commands to **turtle** to perform the drawing. Finally, the last two lines simply keeps the turtle graphics window open until we press enter (in the IDLE window) to close it.

**2. It's not *magic*.**

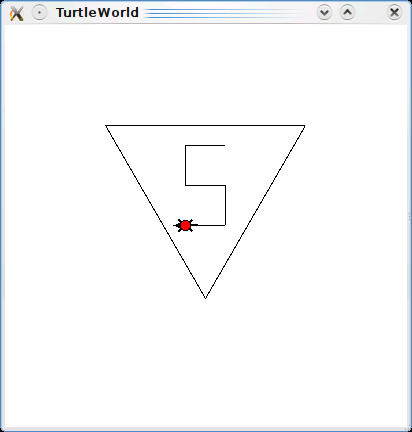
As explained in class with respect to good programming practice, the program has several comments (the lines beginning with **#**). However it contains several *magic numbers* that will make the program both difficult to read and maintain.

* Modify the previous program to remove the magic numbers and instead store them in variables with descriptive names **length** and **angle**. These assignment statements need to be done *before* the variables are used in the drawing commands (and don’t forget to add a comment for the assignment code section).
* Replace the magic numbers in the graphics commands with the appropriate variables (which should make the program more readable).
* Add four more drawing commands to make the turtle draw a square and end up back where it started.
* Save the file as **square.py** (make sure to save it on your network drive) and show the instructor your program executing.

**3. It's a bird, it's a plane, it's *super turtle!***

Now it is time to make things a bit more complicated (i.e. you should do some planning *before* you type any code).

* Using the above program as a template, write a program to produce the following output. The figure should scale based on *a single* **length** variable (although you will probably want to create additional variables that are computed based on **length**). The figure should be centered in the screen with the S roughly centered in the triangle. (Hint: The turtle starts in the center of the screen.) Your program should not have any *magic numbers* and should be commented appropriately.



* Save the file as **superturtle.py**
* Print out and attach a copy of your program to this activity.
* Submit your source file through Marmoset.
  + Open a web browser (e.g. Internet Explorer) 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 **program01**
* Click **Browse…** , navigate to your program directory and select your **superturtle.py** file (do not worry about the instructions for jar and zip files).
* Click **Submit project!**