CMPT 120,
Summer 2016, SFU Burnaby
Instructor: Diana Cukierman

ASSIGNMENT #5 : The Turtle artist (with for loops and functions)

This assignment may be team work (of up 2 people each) – SUBMITTED VIA CANVAS

If you submit as a group you need to join a pre-existing group associated to this assignment (Groups for Turtle assignment). If you submit individually do not join any group. Everyone will also have to submit a file explaining who the team members are and how you distributed your work.

DEADLINE : MONDAY July 4, 11:59 PM

Write a program which does a drawing with turtle built-in functions so that the program:

a. Includes some color, fills in some area with some color, changes the pen color in some parts
b. the turtle does at least three right or left turns
c. the pen changes from being up to down at least a couple times
d. there are at least three for-loops
e. you create at least three void or non-fruitful functions (i.e. not returning any value), where the functions do drawings and/or pen movements, but do not return any values
f. You create at least two functions which have some parameters (they may be productive or void functions, they may be functions that count for the previous requirement)
g. The defined functions are called (used and passed arguments to the parameters as needed)
h. The program generates some random values and uses them (such as random colors)

Organize your program as recommended in class: place the functions definitions first and then place the top or main level. Call the functions from the top/main level or also, functions may call other functions (as long as the called functions are placed before).

The drawings you obtain are completely up to you. Enjoy the creative process!

WHAT TO SUBMIT

1. **Submit the turtle program via Canvas**
   a. You will have to submit this turtle program by the deadline (Monday July 4 11:59 PM) via Canvas (similarly to previous submissions).

b. **If you work as a team** (of up to 2 people) you **need to join a pre-existing group** associated to this assignment (Groups for turtle assignment). Go to People → Groups (Search with the keyword 'Turtle' to easily find a group.

c. **Everyone** has to also submit a separate text file where you include the names and student numbers of both team-mates and you describe how you distributed your work. If you worked on your own you should state this in this file.

d. **Additionally, if you will want to show your drawing in class**, email your program to the instructor with subject : *CMPT 120 Turtle program to show in class*.

EXAMPLES to guide the usage of built-in Turtle functions and the definition of our own functions

1. Check labs exercises and examples from labs weeks #6 and week #7 (A couple files from those labs are also provided with this assignment description)
2. Check the LINKS section in Canvas to Turtle Built-in functions and color coding

CLARIFICATIONS about importing modules (such as turtle) and about having multiple functions defined:

1. In your programs, place the *import* statements of any modules that you may need (such as *import turtle or import turtle as t*), **outside the functions** you define, that is, place them at the "top level" of the program.

   By doing so, you will be able to use the functions in the modules (via the dot notation) using the module names (*turtle* or *t*) everywhere BOTH at the top level and also inside any functions you define (with no need to include as parameter to the functions).

2. Using the "import xxx as" format vs. "import xxx" format is up to you, just to abbreviate typing when using the dot notation.

3. Functions need to be defined (placed in the .oy file, with def) before they are used (called or invoked) . They may be all called from the top level, and only if a function needs to use another function, you may call them from a function.

4. DO NOT CALL a function (for example *funB*) from another function (for example *funA*) just when the function *funA* ends and you want to continue execution in the other (*funB*) function. To accomplish this, the top level (or whichever level needs to execute the two functions) should call the two functions in order, in this example, first *funA* and then *funB*.

End of Assignment #5, Turtle artist