Lab 2

Turtle graphics in Mu - turtle objects and turtle methods

Part 1

Copy the code listed below in to the Mu editor and save your code with the filename “**lastname_lab2_part1.py**”

```
#
import turtle
#
colors = ["red", "purple", "blue", "green", "orange", "yellow"]
#
my_window = turtle.Screen()
#
my_window.bgcolor("black")
#
print(my_window.bgcolor())
#
my_window.title("A cool Turtle pattern")
#
bob = turtle.Turtle()
#
bob.speed(0)
#
print(bob.speed())
#
for x in range(360):
    #
    bob.color(colors[x % 6])
    #
    bob.width(x / 100 + 1)
    #
    bob.forward(x)
    #
    bob.left(59)
```

Study the code and figure out what it is doing. Add your header comment to the top of the file and add inline comments after every # in the program to explain the code's behavior.

To help figure out what is going on, you can use the python REPL as shown next.

In the python REPL, import turtle and then create a turtle object

Type `bob = turtle.Turtle()`

Type `dir(bob)`
This should provide a list of attributes and methods associated with the object `“bob”`

Type `help(bob.color)`
This should provide help on the `color` method associated with the object `"bob"`

You can use these commands in the REPL to quickly(?) find a little help with an unfamiliar object or method. Or, you can just Google it…

Show your commented code to your instructor.

□ ________

SOLUTION CODE IN THE FILE: lab2_part1.py

Part 2 - add another turtle object

Starting with your code from part 1, save it under a new filename "**lastname_lab2_part2.py**"

Add another turtle object, call it "`sue`" to the program. Add another list of colors, "`colors1`" and modify the code to have both "`bob`" and "`sue`" drawing at the same time. But, they should be drawing different designs and using different colors.

Use the Mu debugger to step through the code to help find problems.

Add inline comments for all your new lines of code. Update your header at the top of the code.

Show your working code to your instructor.

□ ________

POSSIBLE SOLUTION CODE IN THE FILE: lab2_part2.py

Part 3 - Create a "screensaver" with turtle graphics.

For Part 3 you are going to simulate a small screensaver program. Here is a possible basic structure for the program. Your code should create at least four turtle objects that you will use to create random patterns in the turtle window. Possible pseudocode:

- Create 4 turtle objects.
- In a loop
- Clear the window (use the .clearscreen() method)
- Set the background to a random color. (create a list and select randomly)
- In another loop
- Set each turtle object shape to one of the standard shapes. (create a list and select randomly)

- Set each turtle color to one from a list of colors (create a list and select randomly)
- Randomly move each turtle to different locations Eg. bob.goto(x,y) with x = -400 to 400, y = -300 to 300
- Use the stamp() method to stamp the current turtle shape. Eg. bob.stamp()

Do not be afraid to experiment with your code! Try an idea. If it works great, if you get an error, try to figure out why and learn from your mistakes.

Refactor your code if necessary. Make sure you have your header information at the top and many many inline comments.

Save your file as **lastname_lab2_part3.py**

- Use the Mu debugger to step through the code to help find problems.

Show your working screensaver to your instructor. □ ________

POSSIBLE SOLUTION CODE IN THE FILE: lab2_part3.py