

Cpts 111 Lab #4

In this lab, you will use the following constructs to solve 3 tasks:

1. IF / ELSE statements [tasks 1, 2, & 3]
2. Turtle Graphics [tasks 2 and 3]

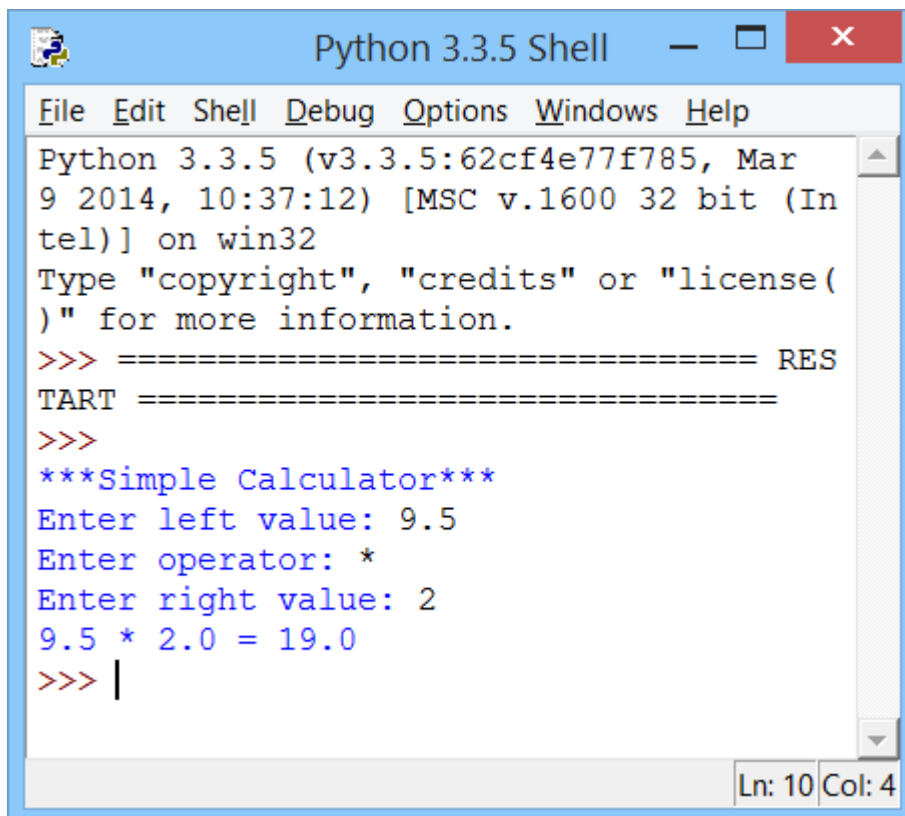
In addition, this lab utilizes the following constructs which are provided for you:

1. WHILE loops

Task 1

In this lab, you will write a simple calculator that can perform addition, subtraction, multiplication, and division. The program flow for your calculator is as follows:

1. Prompt user for left-hand value
2. Prompt user for operator (e.g. +, -, *, /)
3. Prompt user for right-hand value
4. Output result

A screenshot of a Python 3.3.5 Shell window. The window has a blue title bar with the text "Python 3.3.5 Shell" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main area is a text editor showing the following text:

```
Python 3.3.5 (v3.3.5:62cf4e77f785, Mar
9 2014, 10:37:12) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license(
)" for more information.
>>> ===== RES
TART =====
>>>
***Simple Calculator***
Enter left value: 9.5
Enter operator: *
Enter right value: 2
9.5 * 2.0 = 19.0
>>> |
```

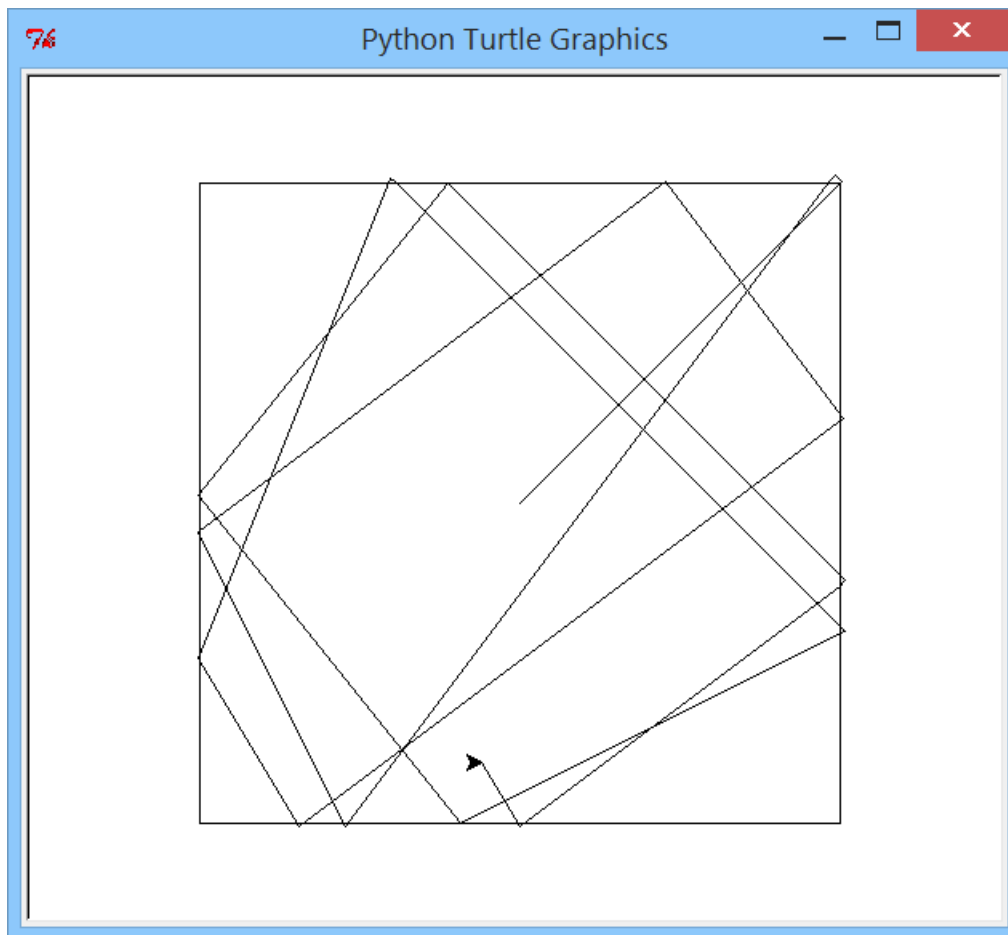
The status bar at the bottom right shows "Ln: 10 Col: 4".

Task 2: Keeping a turtle in-bounds part 1

In this task, you must modify the supplied starter code to keep the turtle within the drawn box. Search the starter code for the "LAB 3 TODO" comment and make your changes there. Note that you can use the `x_position` and `y_position` variables to check to see if the turtle is within the box's boundary (-200 to 200). If the turtle moves outside the boundary, you will need to change one or more of the following variables:

- `y_speed`
- `x_speed`
- `move_x_positive`
- `move_y_positive`

Here's an example of my turtle that is kept in bounds:



Task 3: Keeping a turtle in-bounds part 2

Like the previous task, your goal is to keep the turtle within the bounding box (-200 to 200). However, this time, you can only use the `left()` and `right()` turtle rotation functions to alter the turtle's behavior. Again, search for the "LAB 3 TODO" comment in the supplied task 3 starter code. In order to determine if the turtle has gone out of bounds, you will need to use one or more IF statements based on the variables `"turtle_x"` and `"turtle_y"`, which track the turtle's x and y position. Here's a screenshot of my working program:

