



Reno Technology Academy

Multnomah University Reno/Tahoe

CIS104: Coding in Python

Lesson 3

Readings:

Learning Python

Chapter 3: Module Imports and Reloads, Using exec to Run Module Files pp. 67-75

Chapter 13: while and for Loops

While Loops pp.401-403

Break, continue, pass, and the Loop else pp. 403-409

Chapter 16: Function Basics pp. 491-502

Chapter 17: Scopes pp. 505-542**

*Note: Since we are reading the book out of order, some examples in the reading will have logic that we haven't covered yet. It's ok if you don't know what those are, just extract what the reading is concentrating on and we'll get to the other logic later.

**Note: There is a lot of advanced things in the second half of this chapter. I would like you to read it just so you know about it.

Lab/Homework (10 points)

All homework files can be added to GitHub repository in a folder. After you commit and sync the changes, submit the URL to the folder. I would suggest committing each file when you finish each part. You can sync the commits at the end. Feel free to commit and sync as many times as necessary. A commit/sync doesn't mean the project is finished. I will grade the closest submission that doesn't pass the due date. If you change your submission after the due date and before I grade it, you will receive 50% credit of the difference. For example, the submission before the due date is graded at 70%, but the latest submission grades as a 100%, the final grade will be an 85%.

Calculator:

1. Create file (calculator.py) where you will add functionality for a calculator and a main page (main.py).
2. Main.py will contain the entry point of the program (the main function). Write the code necessary to implement a basic calculator module, by satisfying these requirements:

- a. Use a variable in calculator.py that represents the current value of the calculator.
 - b. Use a variable in calculator.py that represents the memory value of the calculator.
 - c. Write functions that allow you to:
 - i. Clear: Clears the calculator's current value.
 - ii. Add: Adds a number to the calculator's current value and returns the new current value.
 - iii. Subtract: Subtracts a number from the calculator's current value and returns the new current value.
 - iv. Multiply: Multiplies a number to the calculator's current value and returns the new current value.
 - v. Divide: Divides a number from the calculator's current value and returns the new current value.
 - vi. Memory Store: Store the calculator's current value in the memory value.
 - vii. Memory Recall: Change the calculator's current value to whatever is in the memory value and returns the new current value.
 - viii. Memory Clear: Clear the memory value.
 - ix. Invert: Invert the sign of the calculator's current value and returns the new current value.
 - x. Power: Calculates the calculator's current value to a specific power and returns the new current value.
3. Make sure to properly comment all of your functions!
 4. In Main.py, using a while loop, accept input from the user one character/number at a time, call the appropriate calculator function,
 5. Exit the program when the user indicates.