Name: Victoria O’Laughlin
Date: February 20, 2023
Course: IT FDN 110 A

Assignment 06 - Intro to Programming

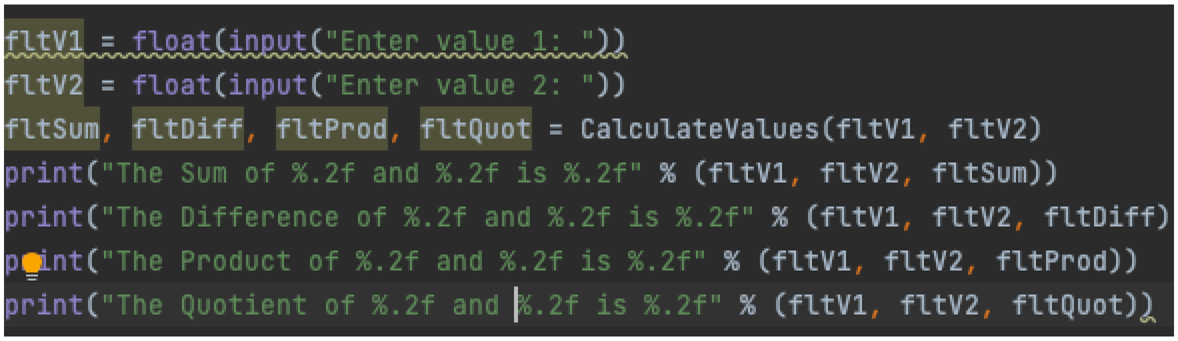
Gitlab Link: https://github.com/vrobado/IntroToProg-Python-Mod06

Introduction:

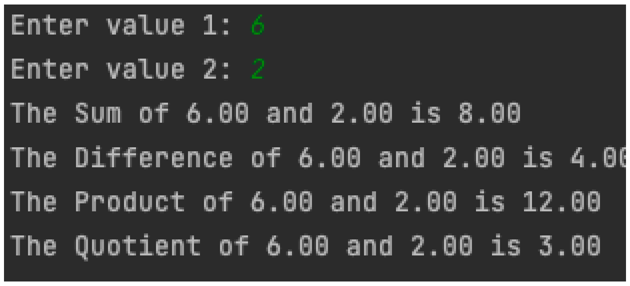
This week introduced the practice of grouping functions together to create classes. This simplifies scripts so that it’s easier to refer to functions by coding name_of_class.name_of_function. The script assignment requires modifying code involving two different classes with multiple functions within them. This document describes my thought process while completing this assignment before summarizing what I learned.

Thought process while writing my script:

1. Labs 6-1 and 6-2 were good introductions of grouping functions together to create classes. Grouping functions together is a ‘nice to have’, but I’ll do it going forward because I like things organized. The labs also introduced parameters, which are a variable used by the function.
2. One thing that had been unknown to me before was how to include variables within strings. The labs had us practice this unknown- the use of %.2f solves it. Refer to Figure 1.0 (two different screenshots for the same figure) for an example of how this is used.
3. Once I finished the labs, the course videos, and other resources, I downloaded Assignment06_Starter and made a copy of it to create Assignment06.py. The first thing I did was run it, and immediately I began getting error messages. I used debug mode to help me understand which line in the code the script wasn’t liking.
4. See Figure 1.1 for an example of an error message that led me to realize I needed to start adding code in my functions - particularly the **input_new_task_and_priority** function.
5. After starting to run into error messages that I couldn’t fully understand, even with debug mode, I went to Canvas and checked out the Assignment 06 Review Video for help. I realized that the hardest part of modifying someone else’s code is needing to invest the time in learning the variables, organization, and patterns that you yourself may not be used to.

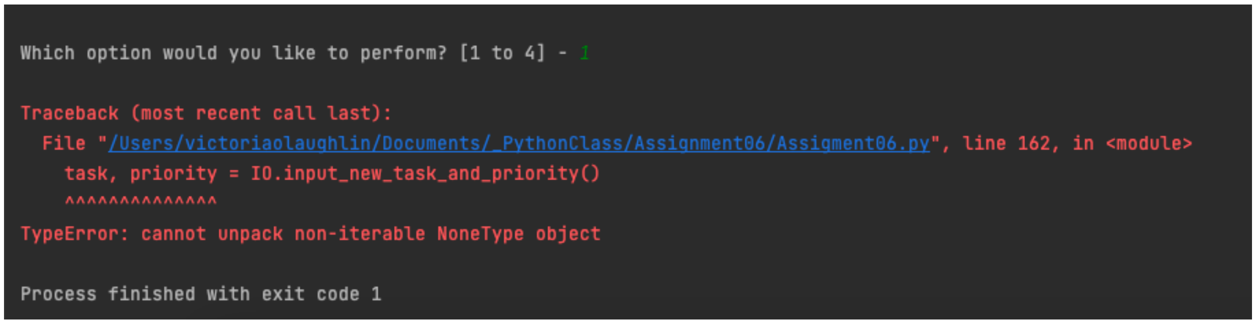


```
fltV1 = float(input("Enter value 1: "))
fltV2 = float(input("Enter value 2: "))
fltSum, fltDiff, fltProd, fltQuot = CalculateValues(fltV1, fltV2)
print("The Sum of %.2f and %.2f is %.2f" % (fltV1, fltV2, fltSum))
print("The Difference of %.2f and %.2f is %.2f" % (fltV1, fltV2, fltDiff)
print("The Product of %.2f and %.2f is %.2f" % (fltV1, fltV2, fltProd))
print("The Quotient of %.2f and %.2f is %.2f" % (fltV1, fltV2, fltQuot))
```



```
Enter value 1: 6
Enter value 2: 2
The Sum of 6.00 and 2.00 is 8.00
The Difference of 6.00 and 2.00 is 4.0
The Product of 6.00 and 2.00 is 12.00
The Quotient of 6.00 and 2.00 is 3.00
```

Figure 1.0: I found the %.2f confusing but helpful.



```
Which option would you like to perform? [1 to 4] - 1

Traceback (most recent call last):
  File "/Users/victoriaolaughlin/Documents/_PythonClass/Assignment06/Assigment06.py", line 162, in <module>
    task, priority = IO.input_new_task_and_priority()
    ^^^^^^^^^^^^^^
TypeError: cannot unpack non-iterable NoneType object

Process finished with exit code 1
```

Figure 1.1: Error message

Summary:

Organization keeps me sane. I enjoy the use of functions within classes to organize my code, in addition to the Separation of Concerns. Going forward, these are two things I'll continue using in my script. Modifying someone else's code is challenging in a different way than writing your own. The biggest challenge for me was tackling unfamiliar error messages. Luckily, I haven't ran into an error message that wasn't Googleable.