Andrew Hopkins
February 6, 2024
IT FDN 110 A
Assignment 06
https://github.com/afhop18/IntroToProg-Python-Mod05

Functions, Classes, and Separation of Concerns

Introduction

This week I learned about how functions and classes work in Python as well as how to organize them according to the separation of concerns. Though there were considerably fewer topics to learn about this week there was a lot more complexity within those topics that I had to wrap my head around in order to make sure I knew what I was doing when I went to implement them. The separation of concerns was the easiest for me to understand as it was a basic guideline to organizing more complex code. Classes were also fairly simple as they acted as containers of functions that I could call on through the class. Functions however were a little bit harder for me to understand, especially when it came to the syntax and use of the parameters and arguments. I did eventually get the hang of them though. I also found it to be a nice change of pace this week that the script assignment wasn't so much about creating code as it was more about reorganizing code that I was already familiar with.

Writing the Script.

Creating an Outline for the Separation of Concerns

Since the assignment this week was more about the organizing code that I had already written into functions, classes, and the separation of concerns rather than writing new code to improve the program the first thing I did was create an outline of the separation of concerns using the classes (one for user experience, and another for file handling) and their respective functions that were required by the assignment. I defined each class and its respective functions with the required parameters and told the script to pass each one in order to properly test them as I pulled their functionality from the "old" main body of the script.

```
class FileProcessor:
    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        pass

    @staticmethod
    def write_data_to_file(file_name: str, student_data: list):
        pass
    pass

class IO:
    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        pass

    @staticmethod
    def output_menu(menu: str):
        pass

    @staticmethod
    def input_menu_choice():
        pass
```

Figure 1: Creating and Outline of the Classes and Functions

Repurposing Original Code as Functions

The next step was to start pulling code from the main body of the script and placing it into the outlined functions. I did this in order of “most used” and then by program flow (menu choice). The lines of code that received the most use was obviously the function to handle exceptions and errors. I also did this one first because I knew that I was going to need to call on that function within other functions as I was reorganizing them. After I finished the output_error_message function I moved on to reading the file, printing the menu, accepting user input for the menu, accepting user input for the registrations, output for the new data, and saving that data to a file, all while ensuring to place each function in the correct class (according to the separation of concerns) and call upon the exceptions function where necessary. Also, I made sure as I was writing each function to provide a header describing the function’s purpose.

```python
class IO:
    """
    This Class contains all the functions that require either input or output by the script
    including error messages / exception handling

    AFH, 2/6/2024, Created Class
    """

    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        """
        This function will display an error message to the user

        AFH, 2/6/2024, Created function

        @param message: accepts a string value
        @param error: accepts a value of type Exception
        @return: No Return
        """
        print("-- Technical Error Message -- ")
        print(message)  # Prints the custom message
        print(error.__doc__)
        print(error.__str__())
```

Figure 2: Example of reorganizing code into a function under a class

Simplifying the Body of the Script

Now that I had organized the bulk of the script into classes and functions I needed to rewrite the main body of the script to perform those functions according to the menu choice selected by the user. The first class.function() command I used was to read in the file because it is the first thing that has to be done and doesn't require a menu choice. I then effectively went from the top down making sure that each menu choice called on the appropriate function by writing the class.function() command in an if / elif / else loop. I have to admit, this was my favorite part of the script to write because of the pure simplicity of it.

```python
# When the program starts, read the file data into a list of dictionaries
# Extract the data from the file
students = FileProcessor.read_data_from_file(file_name=FILE_NAME, student_data=students)

# Present and Process the data
while (True):

    # Present the menu of choices
    IO.output_menu(menu=MENU)
    menu_choice = IO.input_menu_choice()

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        IO.input_student_data(student_data=students)
        continue

    # Present the current data (all active enrollments)
    elif menu_choice == "2":
        IO.output_student_courses(student_data=students)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
        continue
```

Figure 3: Simplifying the main body of the script with class.function() calls

Testing

Testing this script was by far the hardest, and most frustrating part of getting this assignment's program right. The primary reason for the difficulty was getting the parameters and return values correct to make everything function correctly. I'll admit that when I was reading the assignment notes I knew that I wasn't quite grasping the flow of parameters within functions. The labs helped me to understand them a little bit better but I was still a little confused when I was writing out the assignment. Because of this, as I was writing each function and calling on that function in the main body of the script I was debugging every single one of them to verify that the information going in and coming out of it was what I need it to be (the students list of dictionaries gave me the most trouble). There were some moments where I had to go through syntax and variables with a fine toothed comb because something just wasn't right.

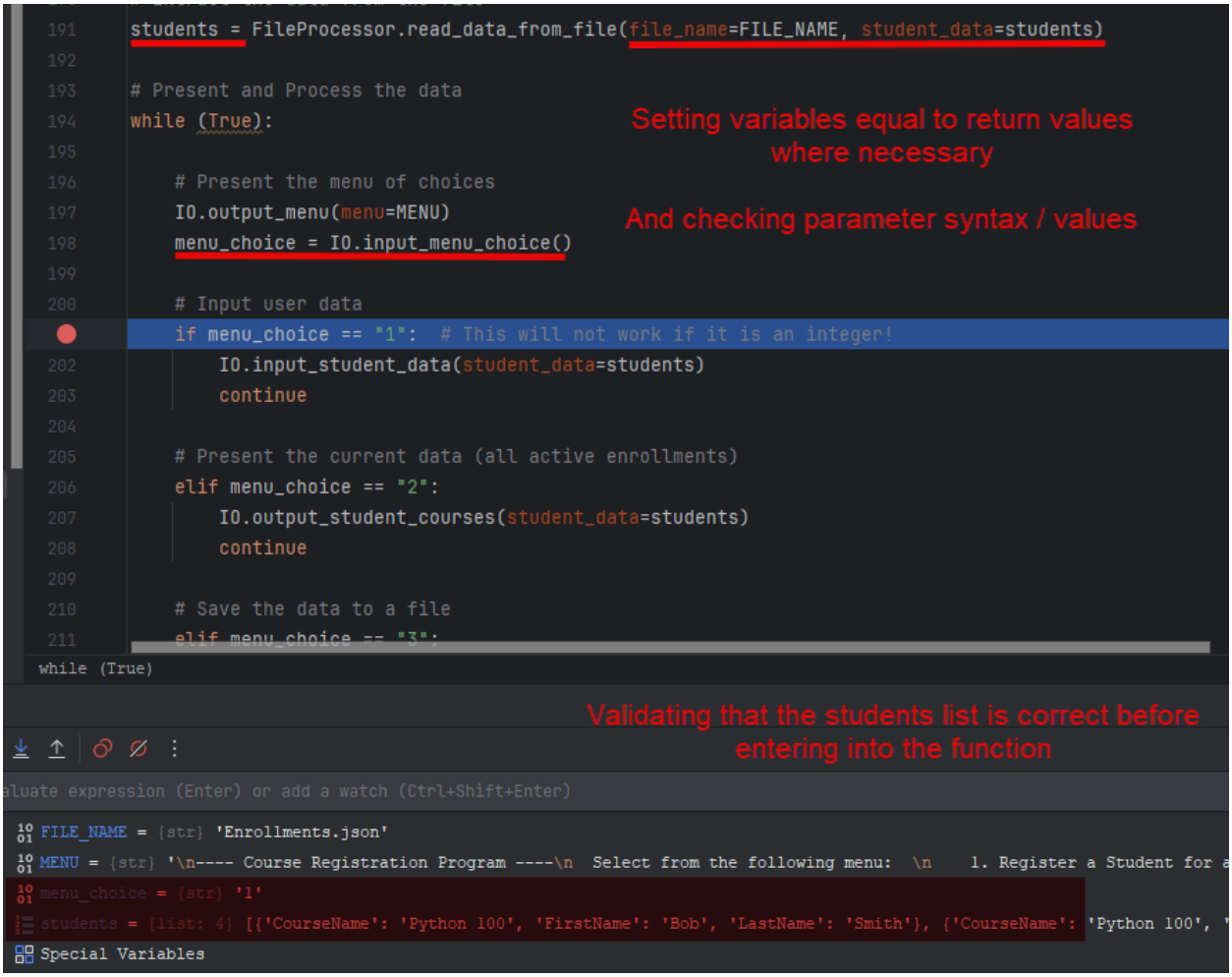

Figure 4: Debugging to ensure correct parameter syntax and variable values

Notes

- There was a moment when I was testing that, because I was doing something wrong with the list of students, that when a function had to iterate over that list with a for-in loop, because it only had one dictionary in the list is stated that the indices had to be of type integer and kept throwing an error which was really strange for me because the exact same code worked once I fixed my issue and the list that it was iterating over had more than one entry.
- There were a few functions I had to create that were totally unnecessary (print menu and menu choice). Turning them into functions meant I had to write dozens of lines of code and comments around what could have been a single line of code in the main body of the script. I understand the purpose of making these functions for the assignment but if I had to choose between that and just putting the single line in the main body, I would choose the latter.

Summary

This assignment was, again, a nice change of direction. I really enjoyed taking something that I had already written and was familiar with and making it simpler by learning about the functions and classes python uses. I also quite enjoy organizing and, as I was finishing module 5 which was becoming more complex, I was wondering if there was an imperative next step to keeping what could be a much longer and more

complex code more readable which is why I was relieved to learn about and use the separation of concern. Even as I finished this assignment my confidence in the parameters and arguments for functions isn't quite at the level I want it to be. I'm going to ensure that in the future when I use them I write them slowly and cautiously so that I can both bolster my understanding of them but also hopefully save myself the headache of having to test them so much.