Name: Alejandro Garcia
Date: May 25, 2024
Class: Python 100
GitHub URL: https://github.com/AlGarCu/IntroToProg-Python-Mod05

Assignment 06

Introduction

This report describes the steps taken to perform Assignment 06 for the Python 001 course, in which the main objective was to put together a Python script that would read from a .json file into a specific data format and provide the user with a menu of options, including collecting specific input to generate output repeatedly. In line with the Module contents, the emphasis was on putting the modularizing and making the code more efficient by applying several functions. Moreover, this task involved uploading the deliverables into a GitHub repository.

Taking stock

I reviewed the materials for Module 06 and took notes where necessary. By far I found the content of this Module to be the most challenging thus far, yet as usual the intuition between the use of Classes, Functions and Software Design were laid out in a very clear-cut fashion.

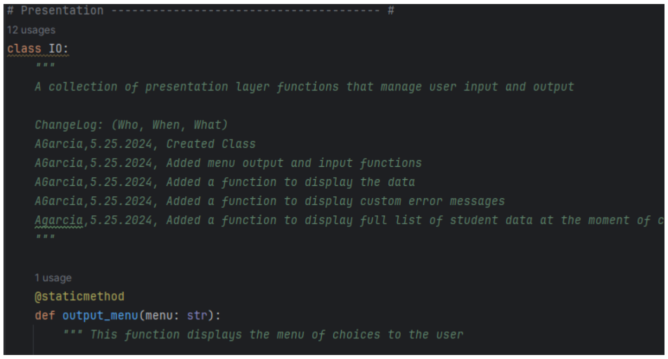
Before moving forward with the script I deep dived into building the feedback from Assignment 5 fully into my baseline code for Assignment 5, though once I got into it I realized it would require a significant step-up from the last deliverable. In particular, being very careful of where the different functions would substitute prior code became a bit challenging on the grounds of taking care of global and local variable definitions, as well as indentation. Once this was done and I felt confident in my ability to respond to all the questions in the “Tip” section of the course materials, I moved forward with the creation of the deliverable script.

Writing the Code

As always, I aimed to observe the programming best practices that have been thus far laid out in the course materials throughout my work.

I first included and updated the script header, and then adjusted the basis file comments and added new ones to *pseudocode* and give my work structure. Now that the code has become more complex and specific chunks are required to carry out different functions, I made more extensive use of comments to *Pseudocode* prior to entering and testing the actual functional script.

I also strived to observe the best practices laid out as far as definition of classes and functions that were a part of Module’s 06 course materials (Figure 1).

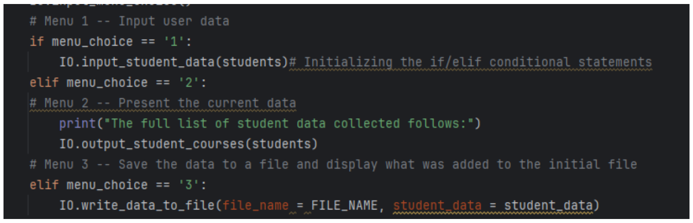


```
# Presentation -------------------------------------- #
12 usages
class IO:
    """
    A collection of presentation layer functions that manage user input and output

    ChangeLog: (Who, When, What)
    AGarcia,5.25.2024, Created Class
    AGarcia,5.25.2024, Added menu output and input functions
    AGarcia,5.25.2024, Added a function to display the data
    AGarcia,5.25.2024, Added a function to display custom error messages
    Agarcia,5.25.2024, Added a function to display full list of student data at the moment of c
    """

    1 usage
    @staticmethod
    def output_menu(menu: str):
        """ This function displays the menu of choices to the user
```

Figure 1: Class and function definitions

Testing

The testing phase itself was fairly similar to what had been done for prior Assignments, yet the code through which such outcomes were accomplished was much more compact as a result of using functions instead of repeated “live code” in the script’s main body. (Figure 2)



```
    # Menu 1 -- Input user data
    if menu_choice == '1':
        IO.input_student_data(students)# Initializing the if/elif conditional statements
    elif menu_choice == '2':
    # Menu 2 -- Present the current data
        print("The full list of student data collected follows:")
        IO.output_student_courses(students)
    # Menu 3 -- Save the data to a file and display what was added to the initial file
    elif menu_choice == '3':
        IO.write_data_to_file(file_name = FILE_NAME, student_data = student_data)
```

Figure 2: Functions make Main Body code much more compact

As in prior occasions, my final check involved ensuring that the output achieved but onscreen and on the json file when run from Pycharm and the console matched.

Summary

This report has sought to describe the steps taken to perform Assignment 06 for the Python 100 course, in which the two key deliverables included a Script that aligned with key programming guidelines emphasizing the contents of Module 6 and accomplished a specific task that centered on Functions, Classes and the Separation-of-Concerns design pattern.