Lim Sia

August 6, 2024
Foundations of Programming: Python
Assignment 06
Selina890/IntroToProg-Python-Mod05: For reviewing homework files (github.com)

Use of functions, classes and separation of concerns pattern

Introduction

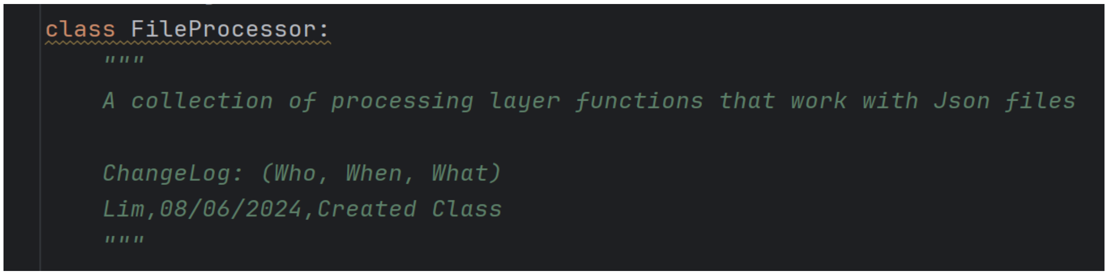
Based on assignment 5, in this assignment, we added three common techniques for improving the scripts using functions, classes, and the separation of concerns programming pattern.

Classes and functions

A function is a reusable block of code that performs a specific task or a set of tasks. The two purposes of functions we focus on, for this assignment, are, (1) modularity, i.e. functions that allow a break down of large programs into smaller, more manageable pieces, so that each function can focus on a specific task, making the code easier to understand, and debug; (2) Reusability, i.e. once a function is defined, its code can be used multiple times throughout the program without having to rewrite the same code. This promotes code reuse and helps prevent redundancy. Classes group functions, variables, and constants by the name of the class. Grouping functions within classes creates a modular structure, making it easier to manage code. Classes are ways to organize code by grouping functions and data needed by those functions, making code more structured and readable, especially in larger projects.

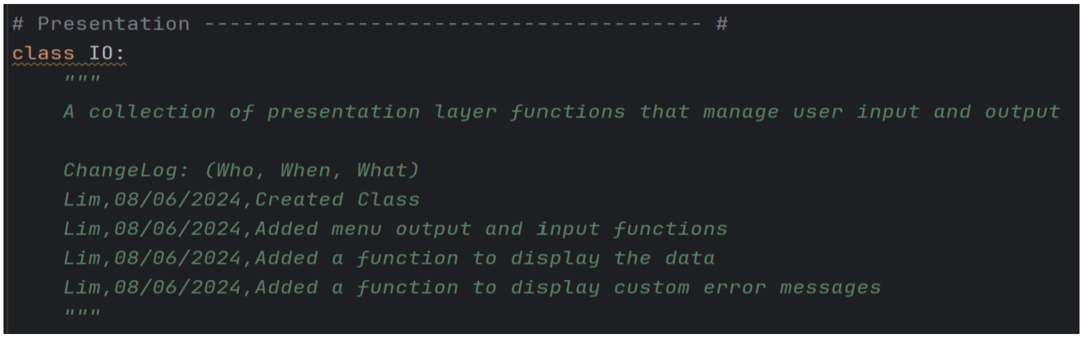
Creating the script using separation of concerns

In terms of separation of concerns, it basically means, for the computer program to be different sections, and each section would be responsible for specific tasks. For this assignment, the various sections include Data, Processing, and Presentation (i.e. Input-Output). To complete assignment 6, the module videos and the assignment review helped a lot, and I copied a lot of codes from Lab 4 in the module. I spent a lot of time understanding and fixing the program errors at the start. Most of the errors involved indent (since I copied from Lab 4), and others involved even punctuation marks, which was hard to locate.



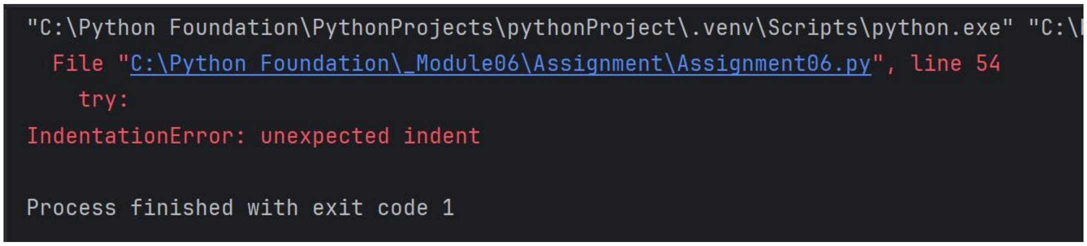
```
class FileProcessor:
    """
    A collection of processing layer functions that work with Json files

    ChangeLog: (Who, When, What)
    Lim,08/06/2024,Created Class
    """
```

Figure 1. Creating a class for file processing



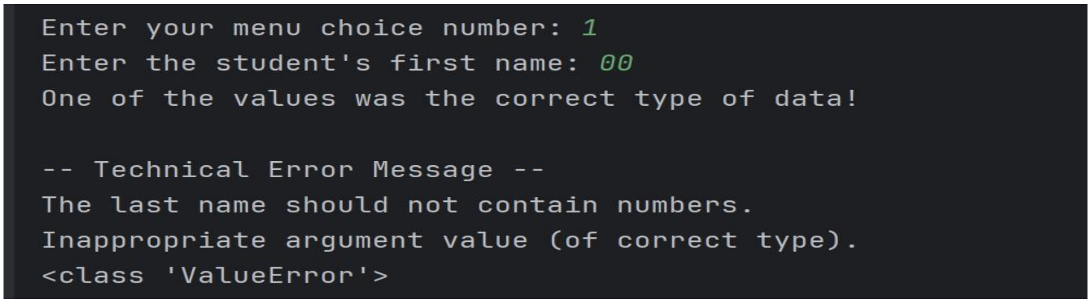
```
# Presentation -------------------------------------- #
class IO:
    """
    A collection of presentation layer functions that manage user input and output

    ChangeLog: (Who, When, What)
    Lim,08/06/2024,Created Class
    Lim,08/06/2024,Added menu output and input functions
    Lim,08/06/2024,Added a function to display the data
    Lim,08/06/2024,Added a function to display custom error messages
    """
```

Figure 2. Creating a class for IO (Input-Output)



```
"C:\Python Foundation\PythonProjects\pythonProject\.venv\Scripts\python.exe" "C:\
  File "C:\Python Foundation\_Module06\Assignment\Assignment06.py", line 54
    try:
IndentationError: unexpected indent

Process finished with exit code 1
```

Figure 3. Error message of indentation

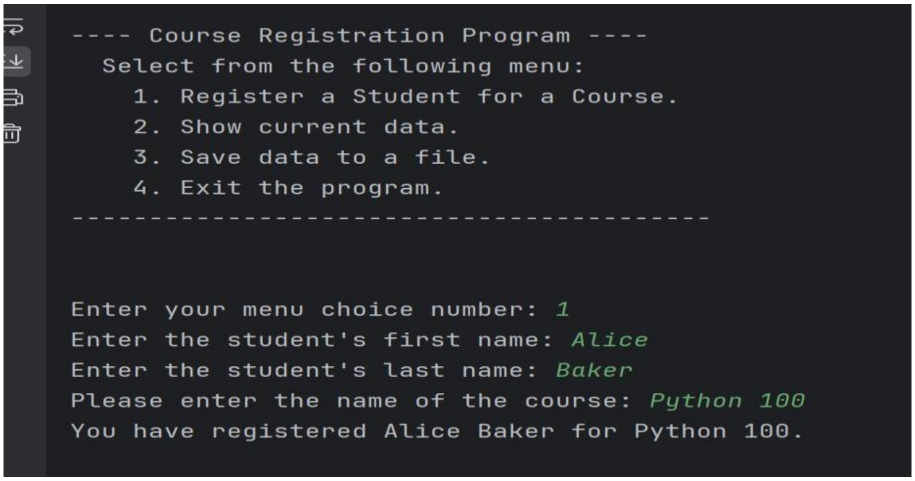


```
Enter your menu choice number: 1
Enter the student's first name: 00
One of the values was the correct type of data!

-- Technical Error Message --
The last name should not contain numbers.
Inappropriate argument value (of correct type).
<class 'ValueError'>
```

Figure 4. Tested the error message when numbers were entered for student's first name

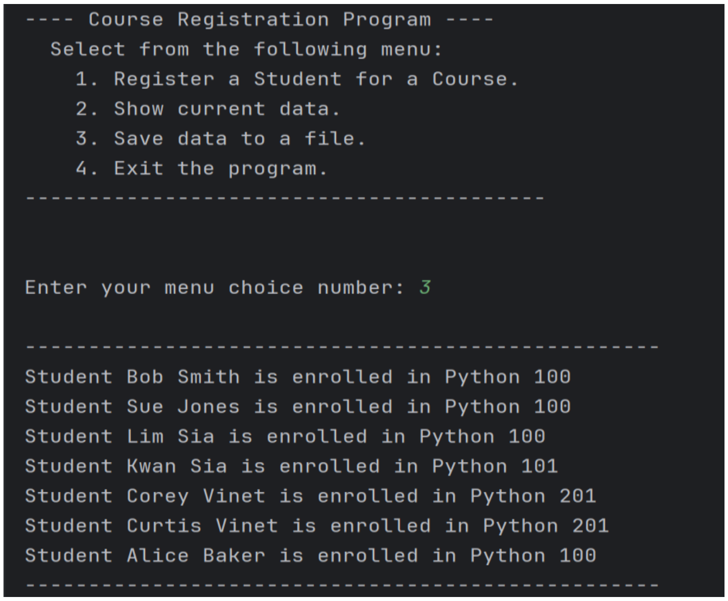
Running the program

After a series of debugging, testing, closing and restarting *PyCharm*, the program finally worked out to be what was designed for it to do.



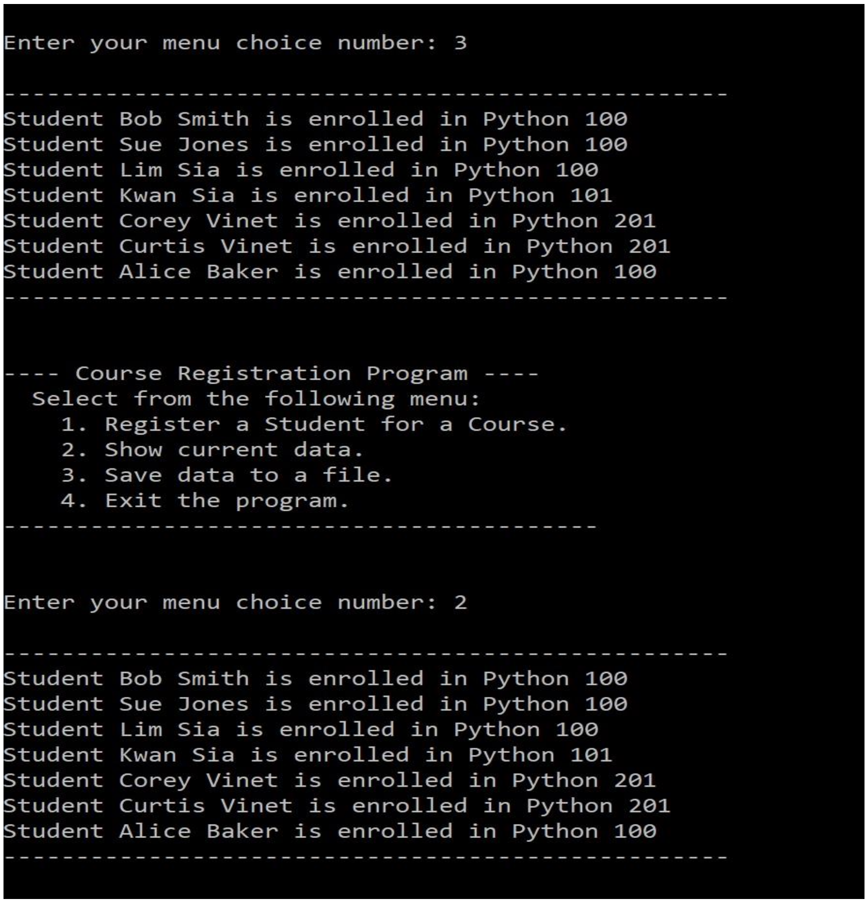
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------


Enter your menu choice number: 1
Enter the student's first name: Alice
Enter the student's last name: Baker
Please enter the name of the course: Python 100
You have registered Alice Baker for Python 100.
```

Figure 5. A student registered to the student list



```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------


Enter your menu choice number: 3

--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Lim Sia is enrolled in Python 100
Student Kwan Sia is enrolled in Python 101
Student Corey Vinet is enrolled in Python 201
Student Curtis Vinet is enrolled in Python 201
Student Alice Baker is enrolled in Python 100
--------------------------------------------------
```

Figure 6. The newly-registered student added to the student list

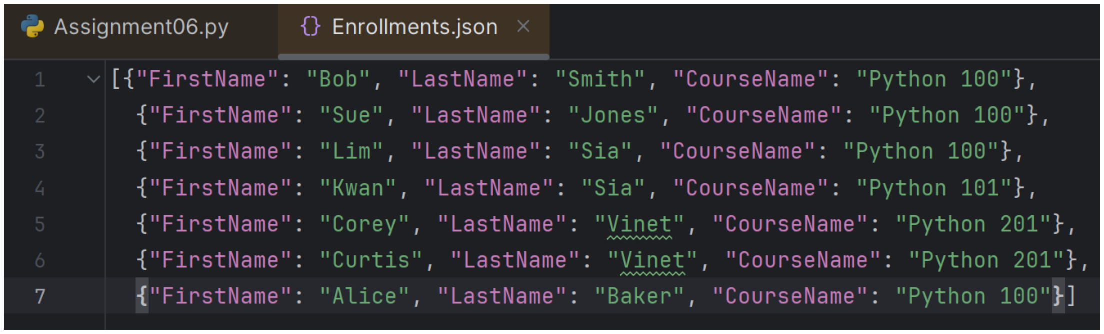


```
Enter your menu choice number: 3

--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Lim Sia is enrolled in Python 100
Student Kwan Sia is enrolled in Python 101
Student Corey Vinet is enrolled in Python 201
Student Curtis Vinet is enrolled in Python 201
Student Alice Baker is enrolled in Python 100
--------------------------------------------------



---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------


Enter your menu choice number: 2

--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Lim Sia is enrolled in Python 100
Student Kwan Sia is enrolled in Python 101
Student Corey Vinet is enrolled in Python 201
Student Curtis Vinet is enrolled in Python 201
Student Alice Baker is enrolled in Python 100
--------------------------------------------------
```

Figure 7. The program runs well on the command prompt

Saving the collected data to the *JSON* file and posting to GitHub

Working with the Json file has been tricky for this assignment, when I had to go back and check on the file all the time in order to make sure that it showed up under the correct directory in order for my program to see the file. Posting to GitHub was straightforward after the assignment 5.



```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"},
 {"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"},
 {"FirstName": "Lim", "LastName": "Sia", "CourseName": "Python 100"},
 {"FirstName": "Kwan", "LastName": "Sia", "CourseName": "Python 101"},
 {"FirstName": "Corey", "LastName": "Vinet", "CourseName": "Python 201"},
 {"FirstName": "Curtis", "LastName": "Vinet", "CourseName": "Python 201"},
 {"FirstName": "Alice", "LastName": "Baker", "CourseName": "Python 100"}]
```

Figure 8. Data shown in the Enrollments.json file

Summary

I spent a lot of time debugging my code, but the assignment notes and the video helped. I also learnt to check on not only the data file, but also punctuation marks in my program, when error messages came through. This assignment impressed me further, that attention to details is very basic and exceptionally important for computer programming, including upper and lower case, space between words, typos, indent and punctuation marks.