Lim Sia

August 1, 2024
Foundations of Programming: Python
Assignment 05

Data processing using dictionaries and exception handling

Introduction

In addition to the loop, conditional logic created using a Python program that demonstrates using constants, variables, and print statements to show messages about a list of students' registrations for Python courses; this assignment requires for us to use data processing based on dictionaries and exception handling, instead of using lists and files as in assignment 4. Besides, instead of saving the data collected onto a csv text file using the *PyCharm IDE*, we are working with a *JSON* file, and we will be posting our assignment to *GitHub*.

JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format that is widely used for data serialization, communication between a client and a server, as well as for configuration files and data storage. *JSON* files consist of key-value pairs, which are very much like Python dictionaries. *JSON* keys are strings enclosed in double quotes, whereas values can be strings, numbers, objects, arrays, booleans, or null. *JSON* data is organized using curly braces {} for objects and square brackets [] for arrays while commas (,) separate key-value pairs or elements within an array.

Creating the script

Based on the scripts in assignment 4, a starter was again provided, where I can add new scripts onto the program and complete assignment 5. The new scripts mainly based on dictionaries instead of lists. While both Lists and Dictionaries in Python are inbuilt data structures that are used to store data, Lists are linear in nature whereas the Dictionaries store data in key-value pairs. It appears a lot easier for me to work with *JSON* rather than *CSV* in terms of coding, however, as a beginner in computer programming, the switch from *CSV* to *JSON* had resulted in confusion, therefore, a lot of time was spent on understanding the program errors at the start.

```
"C:\Python Foundation\PythonProjects\pythonProject\.venv\Scripts\python.exe" "C:\Python Foundation\_M
Traceback (most recent call last):
  File "C:\Python Foundation\_Module05\Assignment\Assignment05.py", line 40, in <module>
    row = {"FistName": data[0], "LastName": data[1], "CourseName": data[2].strip()}
                                           ~~~~^^^
IndexError: list index out of range

Process finished with exit code 1
```

Figure 1. JSON does not work with these codes to read file data

```
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
```

Figure 2. It is straightforward to read JSON files

Error Handing improves the scripts by managing errors. We are required to use "try-except" block to handle errors in this assignment.

```
"C:\Python Foundation\PythonProjects\pythonProject\.venv\Scripts\python.exe" "C:\Python
File not found. Check that the file is in json format
-- Technical Error Message --
File not found.
[Errno 2] No such file or directory: 'Enrollments.json'
```

Figure 3. Error message when the JSON file cannot be located

Running the program

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

```
What would you like to do: 1
Enter the student's first name: Corey
Enter the student's last name: Vinet
Please enter the name of the course: Python 201
You have registered Corey Vinet for Python 201.

---- 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.
----------------------------------------

What would you like to do: 3
The following data was saved to file!
Student Bob Smith is enrolled in Python 101
Student Sue Jones is enrolled in Python 101
Student Lim Sia is enrolled in Python 101
Student Corey Vinet is enrolled in Python 201
```

Figure 4. Adding and saving student registrations

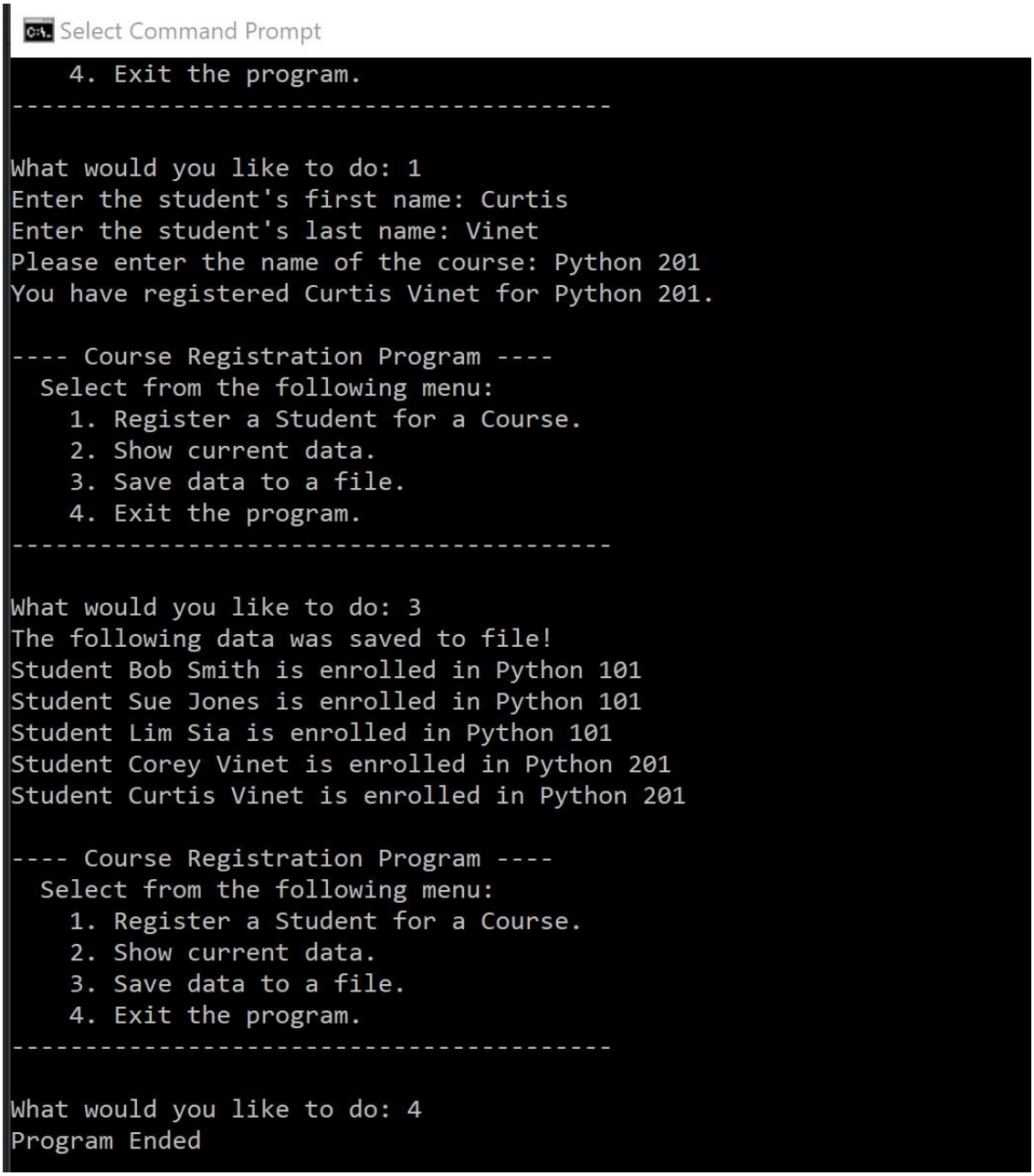

Figure 5. Program runs well on the Command Prompt

Saving the collected data to the *JSON* file

Unlike the previous assignments, when the append mode allows writing to the the file, working with *JSON* is more straightforward.

```
file = open(FILE_NAME, "w")
json.dump(students, file)
```

Figure 6. Writing to JSON file

Figure 7. Data shown in the Enrollments.json file

GitHub

GitHub is often described as a cloud-based platform for code hosting and collaboration. It primarily focuses on code hosting and version control, which encompasses many features that facilitate the sharing and management of code and related files in a cloud-based environment. This makes it a powerful tool for developers and teams to work on software projects collaboratively. To use GitHub, the process is like creating most web software accounts, which is tied to an email account. A repository has to be creased, which is like a set of shared folders where the files are stored and managed through GitHub's web server. When the repository is public, it can easily be seen and shared with other people. We are required to upload assignment 5 onto GitHub.

Summary

I spent a lot of time debugging my codes, but the assignment notes and the video helped. I also learnt to check on the data file in addition to my codes 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, and indent.