

Harpreet Bassi

July 29, 2024

Foundations of Programming, Python

Assignment 05

Advanced Collections of Error Handling

Intro

In assignment 05, I learned about the additional programming tools and techniques using PyCharm. In this document, I will explain the steps I took to create an interactive program similar to assignment 04, where I used my knowledge of Lists, Dictionary, Index, and Key. And finally learned how to implement this data in a JavaScript Object Notation (JSON) file and GitHub.

Creating the Code

I started off by opening PyCharm IDE and Assignment05.py, edited the script header, and then began writing my code. As seen in Figure 1, first thing I started with was adding more information into defining the data constants. I included the menu options as instructed in the Mod05-Assignment instructions, and then went on to also define the data constant File Name, which in this case would be a string value, set to "Enrollments.json". I then added the remaining data variables such as the student first name, last name, and course name (set the string values). I then added the student_data set as a list with the [], which indicates each row of the student data. I also included a table of the student data held, named "students" set to a list value. The reason why I used "list" for these two is because I want to store the data inputted by the user and the "list" data type executes that. I then added the "json_data" set to a string which holds the combined string data. I then included the variable "file" set to "None" which holds a reference to an opened file. The final variable added was "menu_choice" set to a string value which holds the choice the user had made from the menu.

As shown in Figure 1, I then went onto adding "import json" which told the script to import the data to a "json" module and gain access to the different functions and methods with the JSON data. The first thing I did to start off the code was to read the file into a list of lists and extract the data from the file. In my code, I started off with "try:" code which was added as a part of its exception handling. The "try" code would allow me to test a block of code for errors, and once it caught an error, it would then be handled by the "except" block code that I also added. As you can see in figure 2, I told the code to open the file (Enrollments.json) as a read version, and then wrote to read and layout the data from the JSON file. I then started with the "except" block which would indicate the code to start looking to see if there are any errors, and based on the certain errors, it would print the message to the user.

I then went onto presenting and processing the data using the `while(True):` code, and then presenting the menu choices using the `print()` function to print the menu, and finally using the `input()` function for the user to input what they would like to choose from the menu selection provided. Once that was completed, I started listing the code based on each `menu_choice` the user would input. Starting with `menu_choice 1`, I included the `try:` block code, so if there is any error on the users end, the code will display a message. For example, as shown in Figure 2, you can see how I included the original `input()` function for the user, and then stated `if not student_first_name.isalpha():` then `raise ValueError("First name must contain only alphabetical characters.")`. This part of the code indicated that if the input of the student first name was not alphabetical, it will print out the error message that the name must be in alphabetic characters. I then continued the same situation with the student's last name as well. I then went onto creating a Dictionary with the student data inputted by the user with the information of their first name, last name, and course name. I then appended the dictionary I created to a list, and then printed the confirmation message using the f-string of the users first name and last name, and the course they have registered for. I then included the `continue` statement to skip the remaining of the current iteration of a loop and proceed onto the next one. After, went onto handling the exceptions that are raised on the `try` block, and if there was, it would print the exception message of `e` to let the user know of the problem.

I then added the code if the user chooses `menu_choice 2`, as shown in Figure 3. First thing I did was added the code that displays the 50 hyphens to create a visual separator using the `print()` statement (`print("-" * 50)`). I then added the loop that goes over each dictionary in the students list, and then created a `print()` statement that would output the f-string for each of the student information provided. I added another code that displays the 50 hyphens to close it out, and then used `continue` statement to have the loop to skip the remaining code in the current iteration and move onto the next iteration.

As you can see in Figure 3, next thing I did was create the code to save the data to the file that was provided. First thing I did was tell the code that if the user inputted the `menu_choice` of 3, then to open the file in the write view, and add the f-string statement of the user's inputted information, when then I added the code to write to the `json_data` which holds the combined string data, and then added the `indent=1` so the code can be written in the file cleaner with 1 line indent. I also included a `print()` statement that informs the user the information was saved to the file. After that, I included the other `for` loop so it can go over each dictionary in the `'students'` list, and then used the `print()` statement f-string within the loop formats and outputs a string for each student, showing their full information they enrolled in. I then added the `except` block that catches any Exception that may be raised during the execution of the code inside the `try` block. If the exception does occur, I added to the code to print the code `"Unexpected error saving the data to the file"` in a `print()` statement, and then finally used the `continue` statement to have the loop skip the remaining code in the current iteration and move onto the next iteration.

The final thing I did was created the code to stop the code using `menu_choice 4`. Added the `break` statement to break out of the loop, and then the `else` statement so that if option 4 is not chosen, it will print the statement `"Please only choose option 1, 2, or 3"`. And then finally added the last `print()` statement that states to the user `"Program Ended"` if `menu_choice 4` is chosen.

```

1  # ----- #
2  # Title: Assignment05
3  # Desc: This assignment demonstrates using dictionaries, files, and exception handling
4  # Change Log: (Who, When, What)
5  #   Harpreet Bassi, 7/29/2024, Created Script for Assignment05
6  # ----- #
7
8  # Define the Data Constants
9  MENU: str = ''
10 ---- Course Registration Program ----
11     Select from the following menu:
12     1. Register a Student for a Course.
13     2. Show current data.
14     3. Save data to a file.
15     4. Exit the program.
16 -----
17 '''
18 # Define the Data Constants
19 FILE_NAME: str = "Enrollments.json"
20
21 # Define the Data Variables and constants
22 student_first_name: str = '' # Holds the first name of a student entered by the user.
23 student_last_name: str = '' # Holds the last name of a student entered by the user.
24 course_name: str = '' # Holds the name of a course entered by the user.
25 student_data: list = [] # one row of student data
26 students: list = [] # a table of student data
27 json_data: str = '' # Holds combined string data.
28 file = None # Holds a reference to an opened file.
29 menu_choice: str # Hold the choice made by the user.
30
31 import json
32
33 # When the program starts, read the file data into a list of lists (table)
34 # Extract the data from the file
35 try:

```

Figure 1: Screenshot of my Final Script in the PyCharm development program

```

35     try:
36         with open(FILE_NAME, "r") as file:
37             students = json.load(file)
38             for item in students:
39                 print(f"First Name: {item['first_name']}, Last Name: {item['last_name']}, Course: {item['course_name']}")
40 # Include the except statements to show if the following do not happen, state the print() statement assigned
41 except FileNotFoundError:
42     print(f"File {FILE_NAME} not found. Starting with an empty List.")
43 except json.JSONDecodeError:
44     print(f"Error reading {FILE_NAME}. Starting with an empty list.")
45 except Exception as e:
46     print(f"Unexpected error reading {FILE_NAME}: {e}")
47
48 # Present and Process the data
49 while (True):
50
51     # Present the menu of choices
52     print(MENU)
53     menu_choice = input("What would you like to do: ")
54
55     # Input user data
56     if menu_choice == "1": # This will not work if it is an integer!
57         try:
58             student_first_name = input("Enter the student's first name: ")
59             if not student_first_name.isalpha():
60                 raise ValueError("First name must contain only alphabetical characters.")
61             student_last_name = input("Enter the student's last name: ")
62             if not student_last_name.isalpha():
63                 raise ValueError("Last name must contain only alphabetical characters.")
64             course_name = input("Please enter the name of the course: ")
65
66             student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': course_name}
67             students.append(student_data)
68             print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
69             continue

```

Figure 2: Screenshot of my Final Script in the PyCharm development program

```

69             continue
70         except ValueError as e:
71             print(e)
72
73     # Present the current data
74     elif menu_choice == "2":
75
76         # Process the data to create and display a custom message
77         print("-" * 50) #display the 50 hyphens
78         for student in students:
79             print(f"Student {student['first_name']} {student['last_name']} is enrolled in {student['course_name']}")
80         print("-" * 50) #display the 50 hyphens
81         continue
82
83     # Save the data to a file
84     elif menu_choice == "3":
85         try:
86             with open(FILE_NAME, "w") as file:
87                 json.dump(students, file, indent=1) #Used Indent=1 for better printing
88
89             print("The following data was saved to file!")
90
91         except Exception as e:
92             print('Unexpected error saving the data to the file')
93             print(e)
94         continue
95
96     # Stop the loop
97     elif menu_choice == "4":
98         break # out of the loop
99     else:
100         print("Please only choose option 1, 2, or 3")
101
102 print("Program Ended")

```

Figure 3: Screenshot of my Final Script in the PyCharm development program

Troubleshooting

The main troubleshooting I had was when I would save the data to the file, it would not indent the new information added. I started doing research on Google and came across an article (Python PrettyPrint JSON Data, PyNative), that referred to using the Indent parameter which creates a new indentation value since Python does not use indentations on the JSON file and uses single lines instead, which can make it difficult for the reader. I then added the Indent parameter in the “json.dump()”, which then displayed the data more organized and easier to see (as seen in Figure 4).



```
1  [
2    {
3      "first_name": "Harpreet",
4      "last_name": "Bassi",
5      "course_name": "Python 100"
6    },
7    {
8      "first_name": "Sam",
9      "last_name": "Johnson",
10     "course_name": "Python 200"
11   },
12   {
13     "first_name": "John",
14     "last_name": "Vance",
15     "course_name": "Python 100"
16   }
17 ]
```

Figure 4: Screenshot of the Enrollments.json file with “Indent” parameter used.

Save the Script

I created a folder in Documents called “Python” and saved my python script as “Assignment05.py”.

Running the Script in Terminal and PyCharm

I opened the Terminal console on my MacBook and went to the folder where the python script is located using the cd (change directory) command. I started off with using the cd command for Documents. I mainly used this for the Documents for tracking purposes for myself. I then went onto using the cd command for the Python folder in Documents, used the cd command again for the “A05” folder where the script is located, and then went onto using “python3 Assignment05.py” which then directed me to the code. Once I inputted the

information the code prompted me to, I was able to successfully have the program run on both PyCharm (Figure 5-5.5) and Terminal (Figure 6-6.5).

```
/usr/local/bin/python3.12 /Users/bassi/Documents/Python/A05/Assignment05.py
First Name: Harpreet, Last Name: Bassi, Course: Python 100
First Name: Sam, Last Name: Johnson, Course: Python 200
First Name: John, Last Name: Vance, Course: 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.
-----

What would you like to do: 1
Enter the student's first name: Samantha
Enter the student's last name: Martin
Please enter the name of the course: Python 100
You have registered Samantha Martin for 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.
-----

What would you like to do: 2
-----

Student Harpreet Bassi is enrolled in Python 100
Student Sam Johnson is enrolled in Python 200
Student John Vance is enrolled in Python 100
Student Samantha Martin is enrolled in Python 100
```

Figure 5: Screenshot of the final run for Assignment05.py in PyCharm.

```
What would you like to do: 2
-----
Student Harpreet Bassi is enrolled in Python 100
Student Sam Johnson is enrolled in Python 200
Student John Vance is enrolled in Python 100
Student Samantha Martin 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.
-----

What would you like to do: 3
The following data was saved to file!

---- 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: 4
Program Ended

Process finished with exit code 0
```

Figure 5.5: Screenshot of the final run for Assignment05.py in PyCharm.

```
Last login: Wed Jul 31 17:53:42 on ttys000
bassi@Harpreets-Air ~ % cd Documents
bassi@Harpreets-Air Documents % cd Python
bassi@Harpreets-Air Python % cd A05
bassi@Harpreets-Air A05 % python3 Assignment05.py
First Name: Harpreet, Last Name: Bassi, Course: Python 100
First Name: Sam, Last Name: Johnson, Course: Python 200
First Name: John, Last Name: Vance, Course: Python 100
First Name: Samantha, Last Name: Martin, Course: 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.
```

```
-----
What would you like to do: 1
Enter the student's first name: Bob
Enter the student's last name: Baker
Please enter the name of the course: Python 200
You have registered Bob Baker for Python 200.
```

```
---- 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: 2
```

```
-----
Student Harpreet Bassi is enrolled in Python 100
Student Sam Johnson is enrolled in Python 200
Student John Vance is enrolled in Python 100
Student Samantha Martin is enrolled in Python 100
Student Bob Baker is enrolled in Python 200
-----
```

```
---- 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!
```

Figure 6: Screenshot of the commands to locate the folder, script, and run Assignment05.py


```
---- 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: 4
Program Ended
bassi@Harpreets-Air A05 %
```

Figure 6.5: Screenshot of the commands to locate the folder, script, and run Assignment05.py

File Processing

Once I ran the code on PyCharm and Terminal, I then went to my Python folder which would have the csv file saved. As seen in Figure 7, I could see the csv file that was labeled as “Enrollments.csv” just like how I had indicated in my code. I opened the file, which then opened to the TextEdit in Mac, as shown in Figure 8. As you can see in Figure 7 and 8, the code successfully implemented the information in a csv file.





< > A05		⋮ ⬆	
Name		^	Date Modified
 Assignment05.py			Today at 5:41 PM
 Enrollments.json			Today at 5:54 PM

Figure 7: Screenshot of Python folder which includes the csv file.

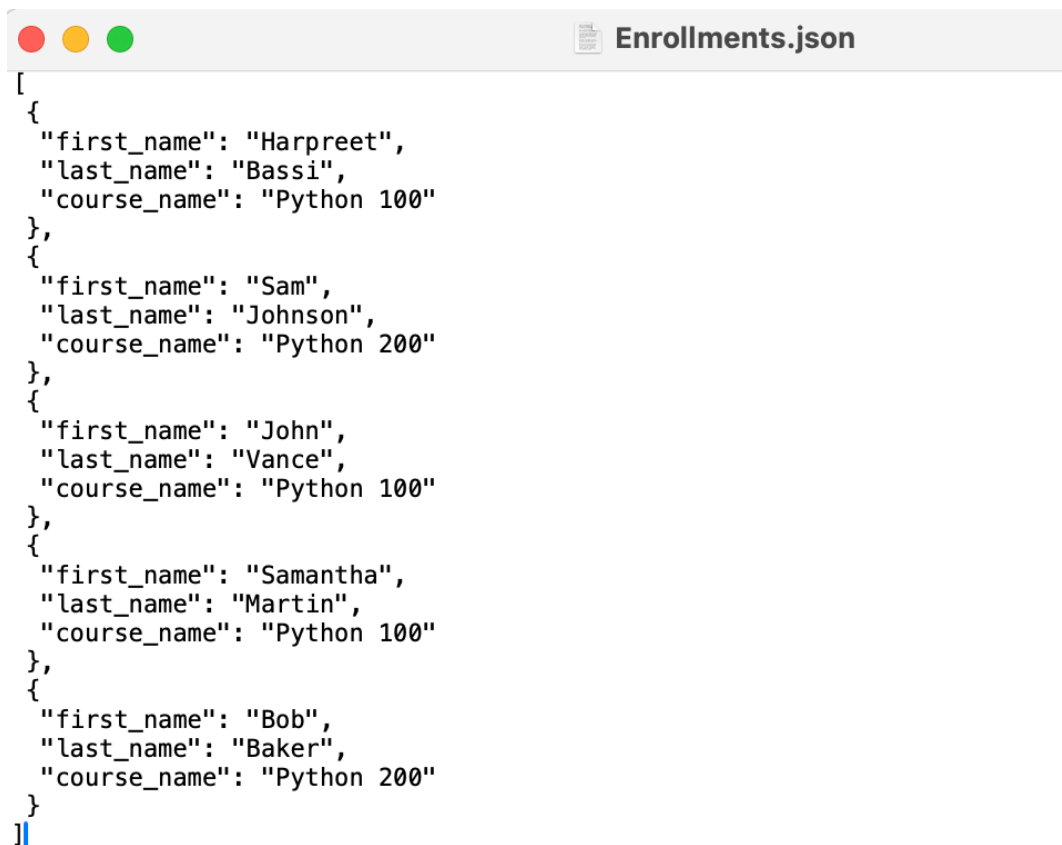


Figure 8: Screenshot of csv file opened in TextEdit on Mac.

Summary

Applying the Module 05 lecture notes and video, I was able to implement and execute an effective Python program that demonstrated my understanding of Lists, Dictionary, Index, and Key. And learned how to implement this data in a JavaScript Object Notation (JSON) file and GitHub. This led me to effectively create a JSON data file of all the information needed regarding a student's registration for a Python course.

References:

- <https://pynative.com/python-prettyprint-json-data/#:~:text=If%20indent%20is%20a%20non,%22%22%20will%20only%20insert%20newlines.>