Mellisa Zheng
May 21, 2025
IT FDN 100 A
Assignment 05
Advanced Collections and Error Handling

Intro

In this assignment, I learned to use more advanced Python tools by working with dictionaries, JSON files, and structured error handling. I completed several guided tasks that helped me extend a course registration program using better data management and stability techniques.

1. Reading and Video Review

I started by reading the Module05 Notes document and watching the required module videos on dictionaries, error handling, and JSON file usage. These resources gave me a solid understanding of how to store structured data using key-value pairs and how to use try-except blocks to catch potential errors in a program. The videos also showed examples of how JSON files can be read from and written to in Python, which helped me implement file storage in my own script.

2. Additional Resources

I also watched four recommended tutorials outside the module: one on using dictionaries in Python, one explaining JSON, one focusing on how to handle exceptions, and one beginner guide to using GitHub. These videos helped clarify how to properly manage data and make my program more reliable and user-friendly. The GitHub tutorial was especially helpful for understanding how to upload files and manage version control.

3. Creating the Registration Script

Planning the Program

Before writing the code, I thought about how to structure the registration system using dictionaries instead of lists, and how to store student records in a JSON file. I planned the program to load existing data when it starts, display a menu using a constant, and allow the user to register a student, view all data, or save the updated list. I also planned where I would use try-except to prevent errors from crashing the program, especially during user input and file access.

Writing the Script

File Name

I named my Python script Assignment05.py as required. This ensures the file is easy to identify and consistent with assignment expectations.

Script Header

At the top of the script, I included a header that shows the title, description, and a change log. I updated the change log with my name and the current date.

Constants

I created two constants in the script:

- MENU: Stores the menu text displayed to the user.
- FILE_NAME: Set to "Enrollments.json" and used throughout the program for loading and saving data.

Both constants are clearly labeled and never modified during the program.

Variables

I defined all required variables at the beginning of the program with type hints:

- student_first_name, student_last_name, and course_name are empty strings.
- file is set to None.
- menu_choice is also an empty string.
- student_data is initialized as an empty dictionary.
- students is an empty list used to store multiple student dictionaries.

Input and Output

Menu Option 1

The program prompts the user to enter a student's first name, last name, and course name. These values are stored in a dictionary named student_data, which is then added to the students list.

Menu Option 2

The program displays the current data by looping through the students list and printing each student's registration information in a readable format.

Menu Option 3

The program writes the data in students to the JSON file using the json.dump() function.

Menu Option 4

The program ends the loop and displays a closing message.

Processing

At the beginning of the script, I added code to load existing student registration data from "Enrollments.json" using json.load(). If the file doesn't exist or is corrupted, the program handles it gracefully and starts with an empty list.

Error Handling

I used try-except blocks in several places:

- To handle file read errors when loading the JSON file.
- To check that the user does not leave first name or last name blank.
- To handle any errors that may occur while writing data to the file.

This makes the program more robust and user-friendly.

Testing

I tested the following:

- Adding multiple students through menu option 1.
- Displaying all current registrations with menu option 2.
- Saving data with menu option 3 and verifying it in the JSON file.
- Reloading the program and seeing the previously saved data.
- Ensuring the program runs correctly in both PyCharm and the terminal.

Source Control

I created a public GitHub repository named IntroToProg-Python-Mod05 and uploaded the following files:

- Assignment05.py
- Enrollments.json with sample data
- KnowledgeDocument.txt (this file)

The link to the GitHub repository is included below for easy access.

GitHub Repository Link: Melissa0407022/IntroToProg-Python: This Rep. will be used

to for reviewing homework files

Summary

This assignment helped me apply dictionaries, JSON file handling, and structured error handling in Python. I also practiced using GitHub to share and store my code professionally. I now feel more confident in organizing data and writing clean, reliable Python programs.