Adnan Dawood

May 15, 2024

Foundations of Programming: Python

Assignment 05

Creating a Python Script

Advanced Collections and Error Handling

Introduction:

This Python program is a course registration system that allows users to register students for courses, view current data, save data to a JSON file, and/or exit the program.

In the world of programming, Python stands out as a versatile and user-friendly language known for its simplicity and readability. This Python program delves into the fundamental concepts of programming - constants, variables, and print statements - to illustrate the process of student registration for a Python course. This program goes beyond the surface to showcase more advanced concepts such as data processing using dictionaries and the importance of exception handling.

At its core, this program provides a practical demonstration of how Python can be utilized to streamline administrative tasks, such as managing student enrollments. By leveraging constants, we establish a foundation of unchanging values, setting the stage for clear and consistent communication throughout the program. Variables, on the other hand, allow for dynamic data storage, empowering us to capture and manipulate information specific to each student's registration details.

Through the strategic use of print statements, we enhance the program's interactivity, providing informative feedback to users at each step of the registration process. From confirming successful registrations to alerting users of potential errors, these print statements serve as a vital communication tool between the program and its users.

The program's sophistication doesn't end there. By employing dictionaries, a versatile data structure in Python, we organize student information in a logical and accessible format. This facilitates efficient data processing, enabling seamless retrieval and manipulation of student data as needed throughout the program's execution.

Furthermore, the program demonstrates a proactive approach to error management through exception handling. By anticipating and gracefully handling potential errors, such as invalid user input or file-related issues, the program ensures smooth execution even in

less-than-ideal circumstances. This emphasis on robustness and reliability underscores Python's reputation as a language that prioritizes user experience and program stability.

In summary, this Python program serves as a practical example of how core programming concepts can be applied to real-world scenarios. By leveraging constants, variables, print statements, dictionaries, and exception handling, it not only showcases the versatility of Python but also empowers users to build more robust and efficient programs in their own projects.

Drafting the Code:

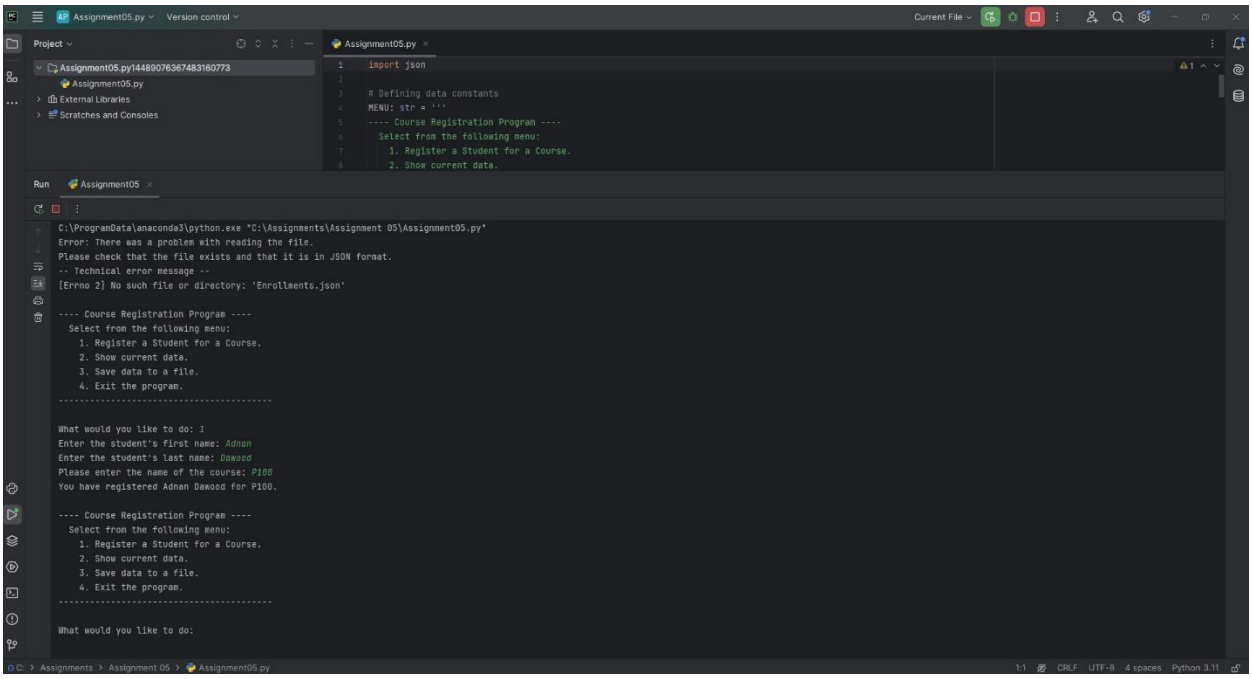
1- Import Required Module: The program begins by importing the json module, which will be used for handling JSON data.
2- Define Data Constants: It defines a multi-line string constant MENU, which presents the options available to the user in the course registration program. It also defines a constant FILE_NAME, specifying the name of the JSON file to store enrollment data.
3- Define Data Variables and Constants: Several variables and constants are defined to store student data, menu choice, file references, etc.
4- Read Data from JSON File (try-except): The program tries to open the JSON file in read mode and load its contents into the students list. If any exception occurs (such as FileNotFoundError or JSONDecodeError), it prints an error message.
5- Main Program Loop (while True): The program enters an infinite loop where it continuously presents the menu options to the user and executes the corresponding actions based on user input.
6- Option 1: Register a Student: If the user chooses option 1, the program prompts the user to input the student's first name, last name, and course name. It validates the input, constructs a dictionary representing the student data, appends it to the students list, and prints a confirmation message.
7- Option 2: Show Current Data: If the user chooses option 2, the program iterates through the students list and prints out each student's first name, last name, and course name.
8- Option 3: Save Data to a File: If the user chooses option 3, the program opens the JSON file in write mode using a context manager (with statement) and writes the contents of the students list to the file using json.dump(). It then prints a confirmation message.
9- Option 4: Exit the Program: If the user chooses option 4, the program breaks out of the loop and ends.
10- Handling Invalid Choices: If the user enters an invalid menu choice, the program prints a message asking the user to choose a valid option.
11- End of Program: Once the user chooses to exit the program, it prints "Program Ended", indicating the end of the execution.

Testing the script and the findings:

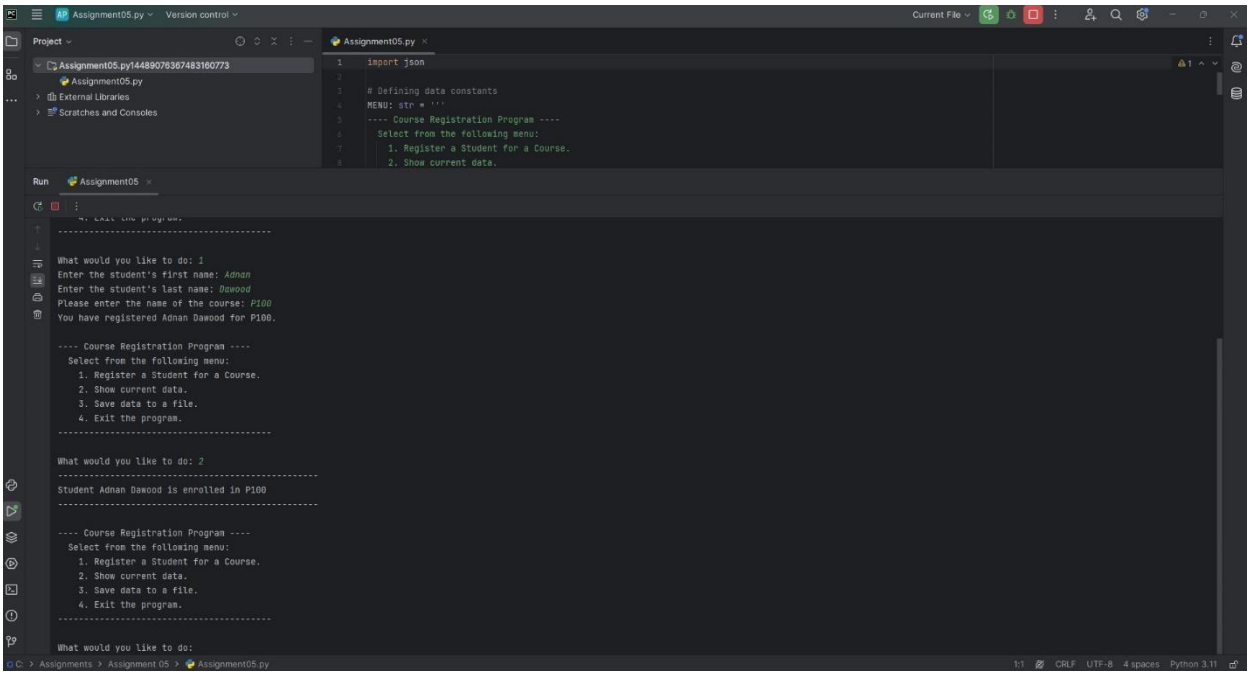
I used PyCharm to evaluate my script. Also, the script was evaluated in terminal.

a- Here is how my script looks like and its output in PyCharm:

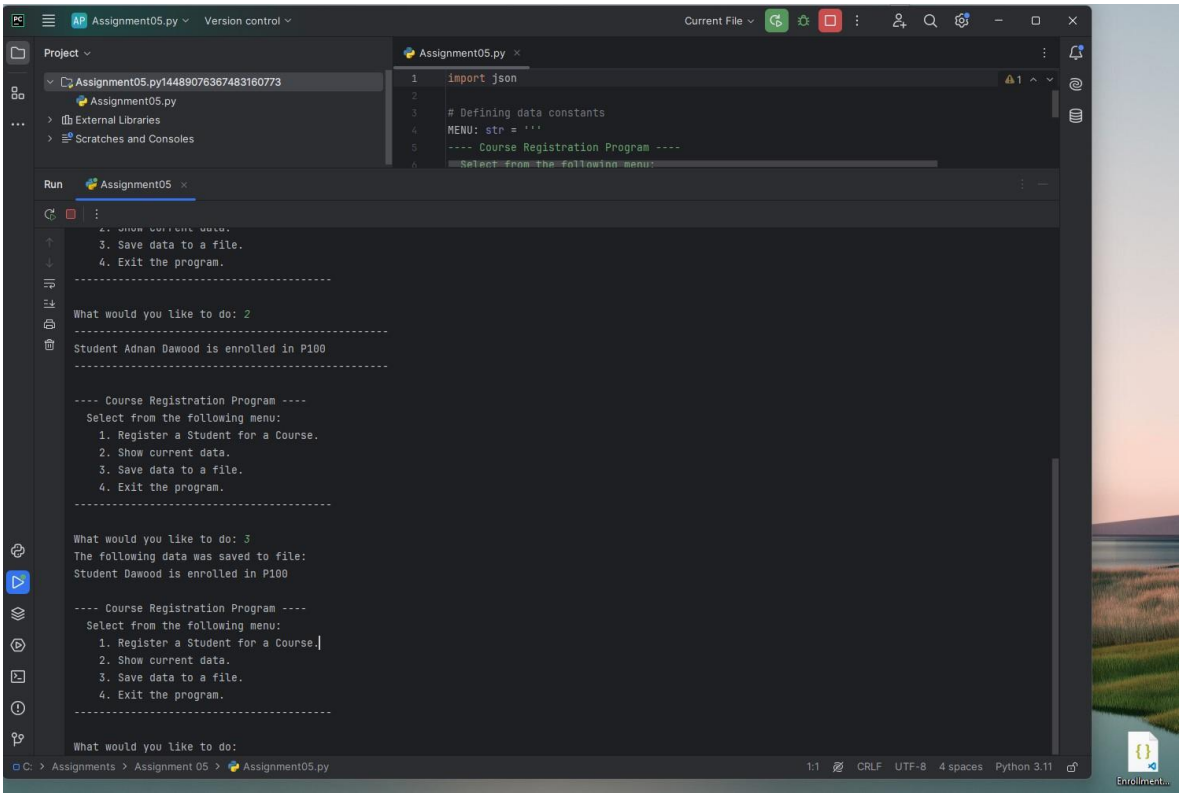
Running the program and entering 1 for option one “Register a Student for a Course” as shown in the picture below:



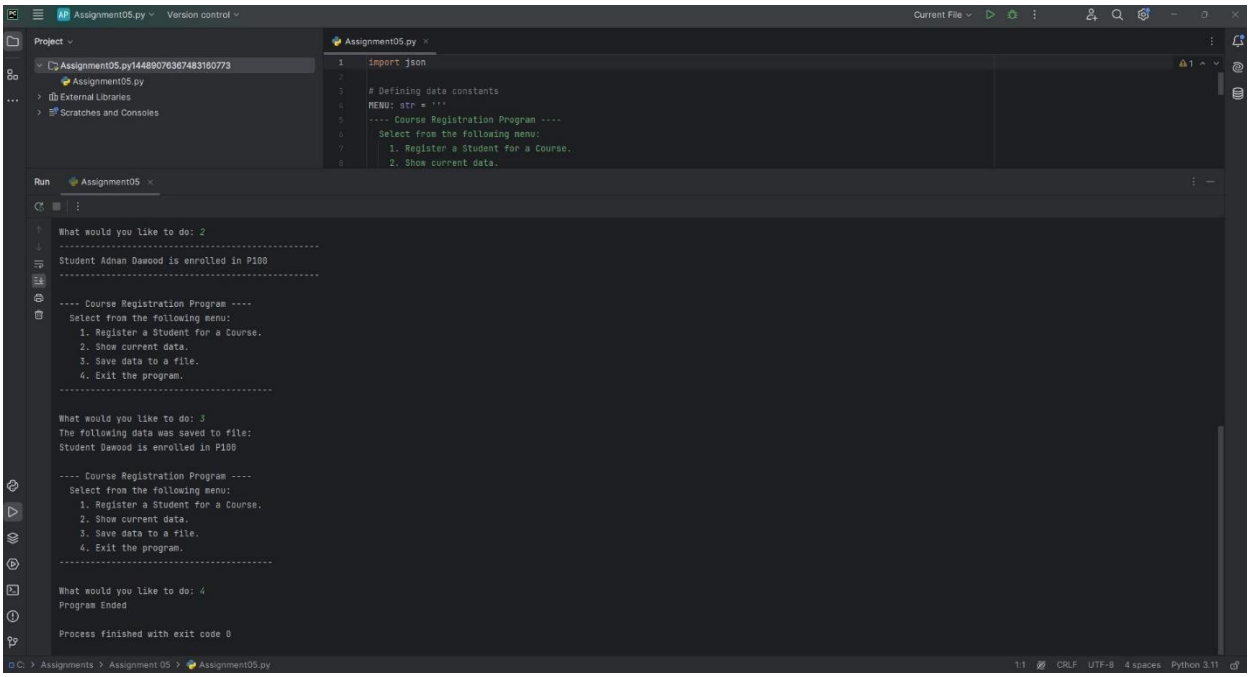
Entering 2 for option two “Show current data” as shown in the picture below:



Entering 3 for option three “Save data to a file (check the right side of the picture)” as shown in the picture below:



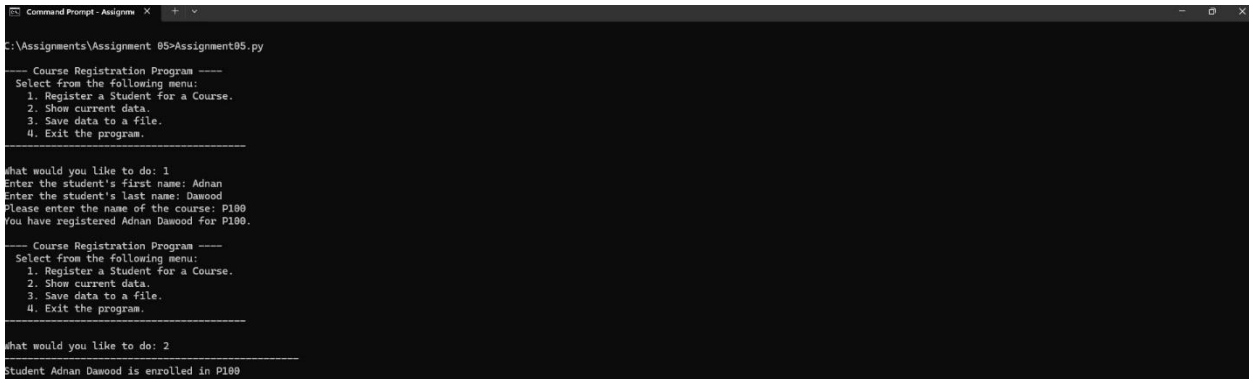
Entering 4 for option four “Exit the program” as shown in the picture below:



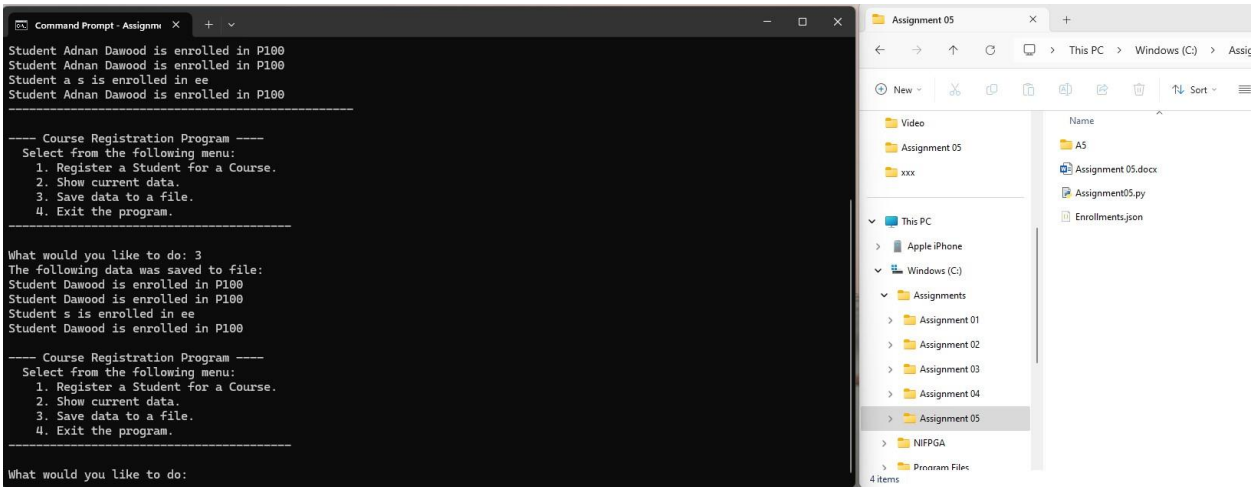
b- Here is how my first script looks like and its output in terminal:

Running the program in terminal and entering 1 for option one “Register a Student for a Course” as shown in the picture below:

Entering 2 for option two “Show current data” as shown in the picture below:



Entering 3 for option three “Save data to a file (check the right side of the picture)” as shown in the picture below:



Entering 4 for option four “Exit the program” as shown in the picture below

Note: This program still cannot verify if the input data is duplicated or not (i.e. it can add/ append the same student information to the json file multiple times).

Summary:

This Python program showcases the use of constants, variables, print statements, dictionaries, and exception handling to manage a student's registration for a Python course. Here's a summary of its key components:

Constants and Variables: The program defines constants like the course name and variables to store student information such as first name, last name, and course name.

Print Statements: Print statements are utilized to display messages to the user, providing information about the course registration process and any errors encountered.

Data Processing with Dictionaries: The student's information is stored in a dictionary data structure. This allows for easy organization and retrieval of student data, with keys representing attributes such as first name, last name, and course name.

Exception Handling: The program incorporates exception handling to manage errors gracefully. For instance, it verifies that the input for the student's name contains only alphabetic characters and raises a ValueError if not. Additionally, it handles potential errors when reading or writing data to a file, providing informative error messages to the user.

By combining these elements, the program provides a robust and user-friendly experience for managing student registrations for a Python course while ensuring data integrity and error resilience.