

Name: Julianne Tillmann

Date: November 21, 2024

Course: IT FDN 110 A Au 24: Foundations Of Programming: Python

Assignment 06

## **Creating a Python Script with Functions, Classes, and Structured Error Handling**

### **Introduction**

The sixth assignment in this course focuses on using functions, classes, and structured error handling to create a robust and user-friendly course registration program. This program allows users to register students for courses, display current enrollments, and save data to a JSON file for persistent storage. Structured error handling ensures the program handles unexpected inputs and missing files. Functions are used to better organize and maintain code.

In creating this script, I worked from the provided starter file and made the necessary changes to meet the requirements. In addition, the latest version of the script is available on GitHub at [IntroToProg-Python-Mod06/ at main · DrJT2013/IntroToProg-Python-Mod06](https://github.com/DrJT2013/IntroToProg-Python-Mod06/).

### **Creating the Script and Adding the Header**

As with all assignments, I began by updating the script header with my name, date, and a brief description of the changes made. This ensures proper documentation and version control.

### **Creating the Main Body of the Script**

#### ***Defining Constants and Variables***

The script defines the following constants and variables:

- MENU: Displays the menu options to the user.
- FILE\_NAME: Holds the name of the JSON file used for data storage.
- students: A list to hold all student records.

#### ***Adding Classes and Methods***

The script is organized using two main classes:

- FileProcessor: Handles reading and writing data to and from the JSON file.
- IO: Manages user input and output, including menu interactions and data entry.

Each class is designed with static methods to maintain modularity and separation of concerns.

#### ***Error Handling***

Structured error handling is implemented using try-except blocks:

- File Not Found: Creates a new file if the JSON file does not exist.

- JSON Decode Error: Handles cases where the JSON file is malformed.
- Input Validation: Ensures that student names only contain letters and that missing data is identified.

### ***Main Features of the Program***

The program offers the following features:

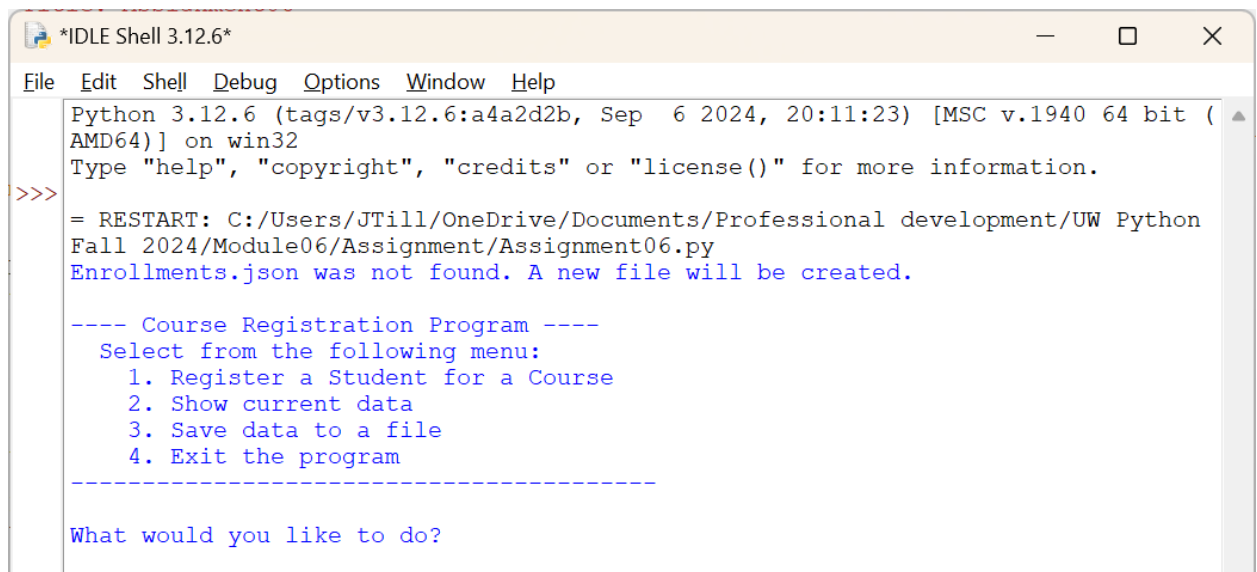
- Menu Option 1: Registers a new student by prompting the user for their first name, last name, and course name.
- Menu Option 2: Displays all registered students and their courses.
- Menu Option 3: Saves the student data to a JSON file.
- Menu Option 4: Exits the program.

Each menu option is implemented with appropriate error handling to ensure the program remains functional in case of errors.

### **Testing the Script**

The program was tested under the following scenarios:

- Starting Without a File: Successfully created a new file and notified the user (Figure 1).



```

Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/JTill/OneDrive/Documents/Professional development/UW Python Fall 2024/Module06/Assignment/Assignment06.py
Enrollments.json was not found. A new file will be created.

---- 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?

```

*Figure 1.* Running the script without a pre-existing file. A new file was created.

- Adding New Students: Registered students with valid inputs and rejected invalid inputs (Figures 2 and 3).

```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help
Fall 2024/Module06/Assignment/Assignment06.py
Enrollments.json was not found. A new file will be created.

---- 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: Ross
Enter the course name: Art 300
Student Bob Ross registered for Art 300.

---- 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
-----

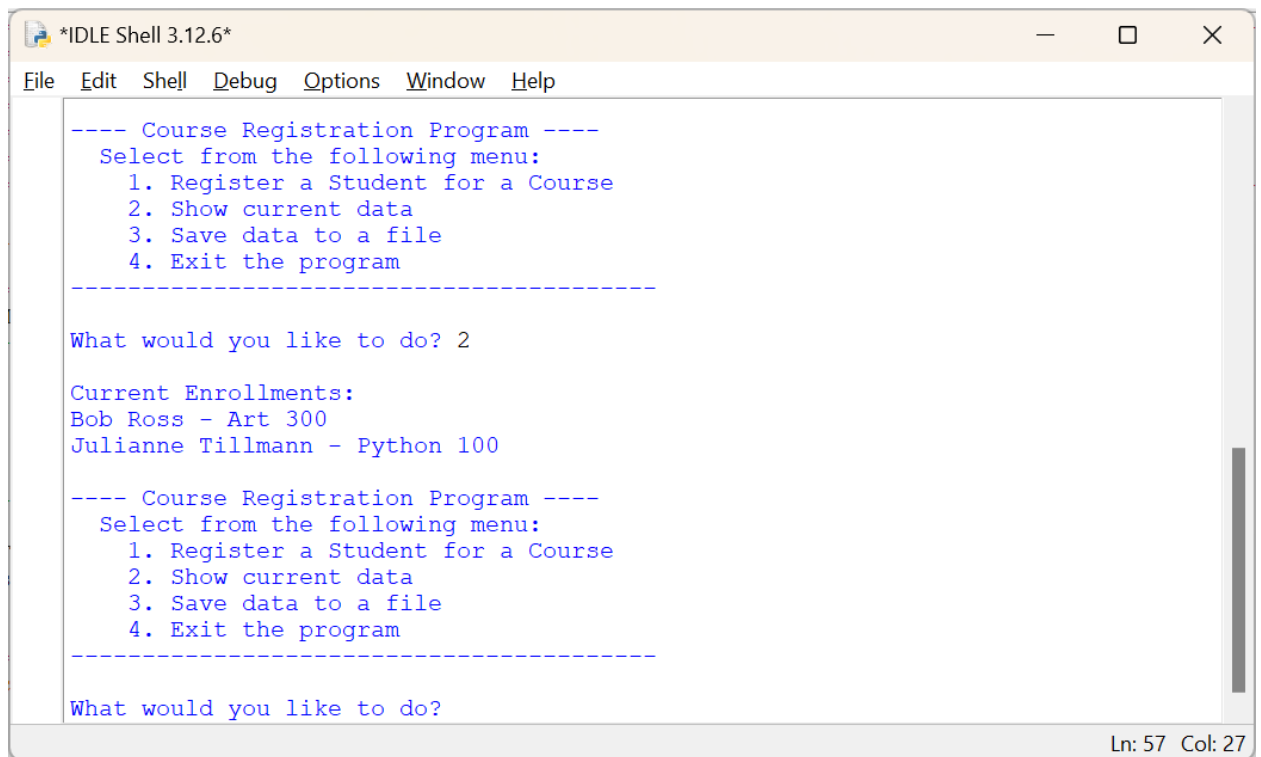
What would you like to do? 1
Enter the student's first name: Julianne
Enter the student's last name: Tillmann
Enter the course name: Python 100
Student Julianne Tillmann registered for Python 100.

---- 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?
```

Ln: 43 Col: 27

Figure 2. Registering new students.



```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help

---- 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

Current Enrollments:
Bob Ross - Art 300
Julianne Tillmann - Python 100

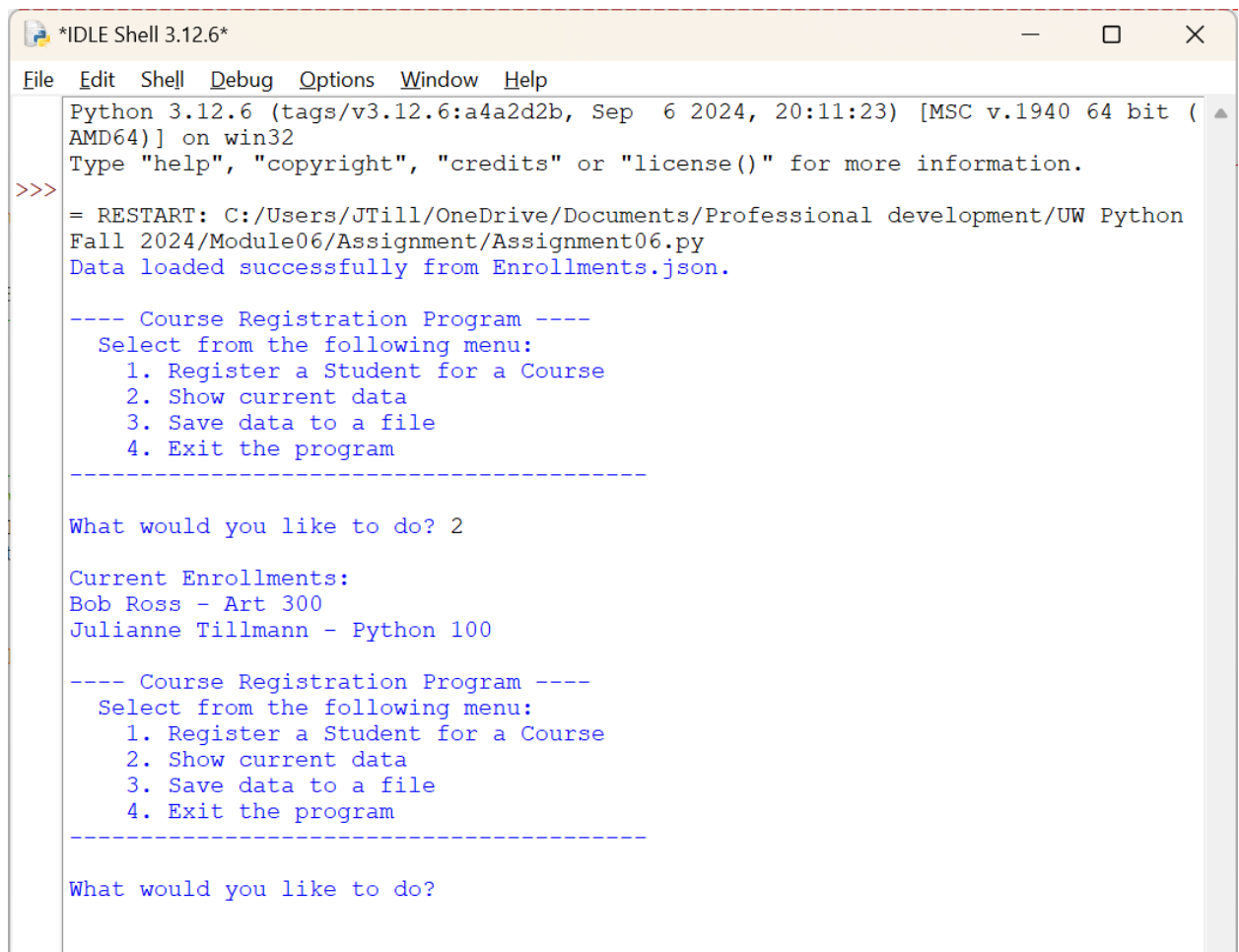
---- Course Registration Program ----
Select from the following menu:
  1. Register a Student for a Course
  2. Show current data
  3. Save data to a file
  4. Exit the program
-----

What would you like to do?
```

Ln: 57 Col: 27

*Figure 3. Reviewing the data with new students added.*

- Reloading with Pre-Existing Data: Loaded data from the JSON file and displayed it accurately (see Figure 4).



```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/JTill/OneDrive/Documents/Professional development/UW Python
Fall 2024/Module06/Assignment/Assignment06.py
Data loaded successfully from Enrollments.json.

---- 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

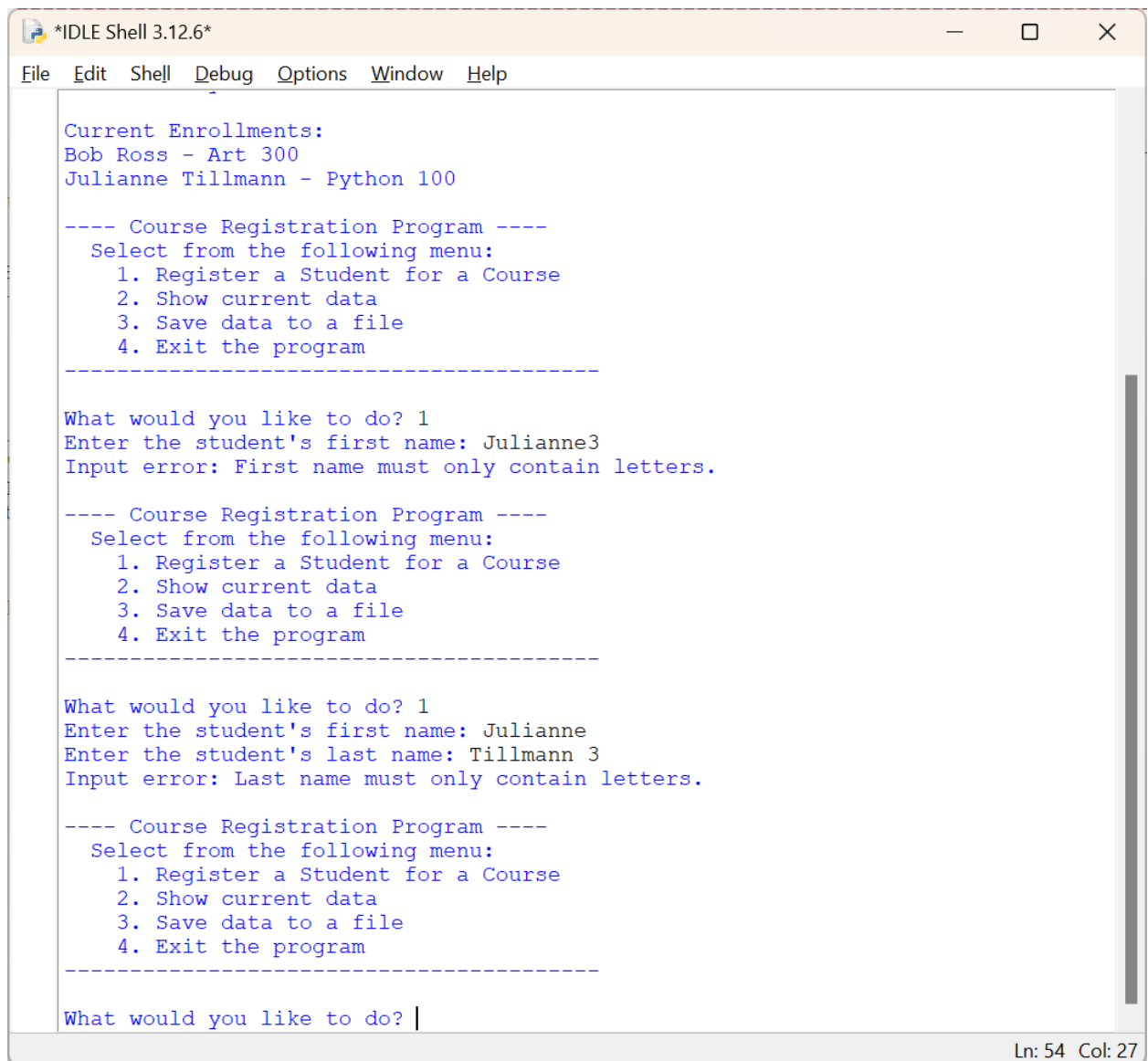
Current Enrollments:
Bob Ross - Art 300
Julianne Tillmann - Python 100

---- 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?
```

Figure 4. Ran the script with a pre-existing file and checked the data.

- Handling Invalid Inputs: Prompted the user to correct errors and prevented the program from crashing (Figure 5).



```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help

Current Enrollments:
Bob Ross - Art 300
Julianne Tillmann - 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: Julianne3
Input error: First name must only contain letters.

---- 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: Julianne
Enter the student's last name: Tillmann 3
Input error: Last name must only contain letters.

---- 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? |
```

Ln: 54 Col: 27

Figure 5. Trying to enter first and last names with non-alphanumeric characters.

The program's robust error handling ensured that all scenarios were handled gracefully, and data integrity was maintained.

## Summary

This assignment enhanced my understanding of Python functions, classes, and error handling. By integrating these features into the course registration program, I improved its functionality and user experience. Testing the program thoroughly allowed me to identify and fix issues, ensuring the program performs reliably in various scenarios.