

Assignment06

Norm De Asis

Foundations Of Programming: Python

University of Washington

Instructor: Randal Root

March 4, 2025

<https://github.com/benchman214/IntroToProg-Python-Mod06.git>

## Purpose

To create a Python program that demonstrates using constants, variables, and print statements to display a message about a student's registration for a Python course. This program is very similar to Assignment02, but **it adds on the while loop, programming menus, conditional logic, and using the PyCharm IDE.**

## Steps and/or Guidance

### Define the requirements:

- The script needed to provide options for registering students, displaying stored data, saving to a file, and exiting
- Data should be stored persistently using a JSON file
- Error handling should be implemented for robustness

### Constants and Variables:

```
Constants for the MENU: str = ""  
FILE_NAME: str = "Enrollments.json"
```

### Declared variables to store student details and handle user input.

```
student_first_name: str = " # Holds the first name of a student entered by the user.  
student_last_name: str = " # Holds the last name of a student entered by the user.  
course_name: str = " # Holds the name of a course entered by the user.  
student_data: dict = {} # one row of student data  
students: list = [] # a table of student data  
csv_data: str = " # Holds combined string data separated by a comma.  
json_data: str = " # Holds combined string data in a json format.  
file = None # Holds a reference to an opened file.
```

### Classes:

- The program includes a class named FileProcessor
- The program includes a class named IO.
- All classes include descriptive document strings

### Data Handling Functions:

```
load_data(): Reads existing student data from the JSON file and loads it into memory  
save_data(): Writes the current student list to the JSON file  
display_data(): Formats and prints the stored student data
```

### Functions:

- All functions include descriptive document strings.
- All functions with except blocks include calls to the function handling error messages.
- All functions use the @staticmethod decorator.
- The program includes functions with the following names and parameters:
  - output\_error\_messages(message: str, error: Exception = None)

- output\_menu(menu: str)
- input\_menu\_choice()
- output\_student\_courses(student\_data: list)
- input\_student\_data(student\_data: list)
- read\_data\_from\_file(file\_name: str, student\_data: list):
- write\_data\_to\_file(file\_name: str, student\_data: list):

### Input / Output:

- On menu choice 1, the program prompts the user to enter the student's first name and last name, followed by the course name, using the input() function and stores the inputs in the respective variables.
- On menu choice 2, the presents a string by formatting the collected data using the print() function.
- Data collected for menu choice 1 is added to a two-dimensional list table (list of dictionaries).
- All data in the list is displayed when menu choice 2 is used.

### Processing

- When the program starts, the contents of the "Enrollments.json" are automatically read into a two-dimensional list table (a list of dictionary rows). (**Tip: Make sure to put some starting data into the file or you will get an error!**)
- On menu choice 3, the program opens a file named "Enrollments.json" in write mode using the open () function. It writes the content of the student's variable to the file using the dump () function, then file is closed using the close () method. Then displays what was stored in the file.

On menu choice 4, the program ends.

### Building User Interaction:

Implemented a loop to present a menu for user selection

Handled input validation for student registration

Processed menu choices to register students, display data, and save to a file

### Error Handling Implementation:

Exception handling for missing or corrupt JSON files

Validation checks to prevent empty names

Handling file writing errors to avoid data loss

- The program provides structured error handling when the file is read into the list of dictionary rows
- The program provides structured error handling when the user enters a first name
- The program provides structured error handling when the user enters a last name
- The program provides structured error handling when the dictionary rows are written to the file

**Testing and Validation:**

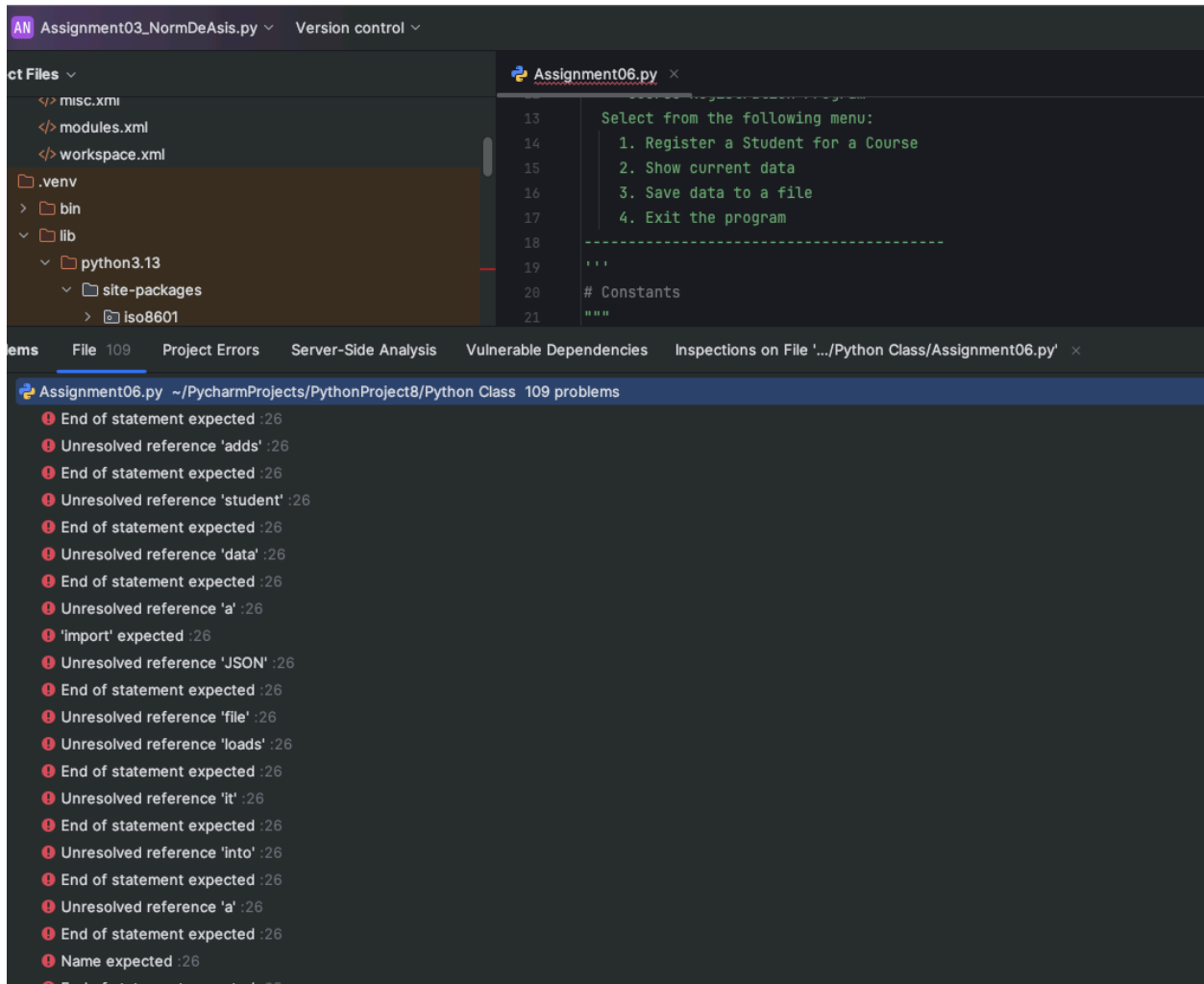
Tested user inputs for multiple scenarios

Verified data persistence by checking saved JSON files

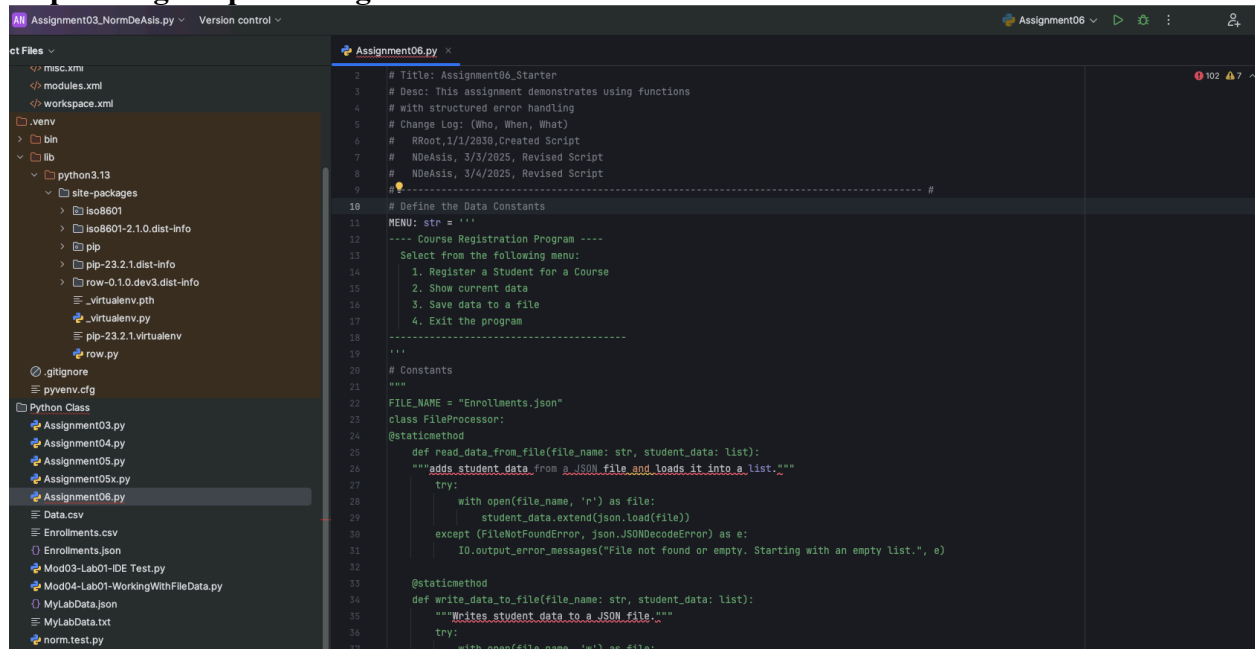
Ensured error handling functions correctly when improper inputs are given

## Lessons Learned

### Too much syntax error



# Duplicating the presenting data



```
1 # Title: Assignment06_Starter
2 # Desc: This assignment demonstrates using functions
3 # with structured error handling
4 # Change Log: (Who, When, What)
5 # RRoot, 1/1/2030, Created Script
6 # NDeAsis, 3/3/2025, Revised Script
7 # NDeAsis, 3/4/2025, Revised Script
8 #
9 #-----#
10 # Define the Data Constants
11
12 MENU: str = '''
13 ---- Course Registration Program ----
14 Select from the following menu:
15 1. Register a Student for a Course
16 2. Show current data
17 3. Save data to a file
18 4. Exit the program
19 -----
20 '''
21 # Constants
22 FILE_NAME = "Enrollments.json"
23 class FileProcessor:
24     @staticmethod
25     def read_data_from_file(file_name: str, student_data: list):
26         """adds student data from a JSON file and loads it into a list."""
27         try:
28             with open(file_name, 'r') as file:
29                 student_data.extend(json.load(file))
30         except (FileNotFoundError, json.JSONDecodeError) as e:
31             IO.output_error_messages("File not found or empty. Starting with an empty list.", e)
32
33     @staticmethod
34     def write_data_to_file(file_name: str, student_data: list):
35         """Writes student data to a JSON file."""
36         try:
37             with open(file_name, 'w') as file:
```

## **References:**

Class Module 06 materials and videos.