September 4th, 2024

IT FDN 110B: Foundations of Programming: Python

Assignment 06

Professor: Luis Conejo

Student: Eder Martins

Creating a Python Script for Menu-Driven User Choices using Classes and Functions

Introduction

The goal of Assignment 06 was to create a Python program using PyCharm that presents the user with a menu from which he/she can choose to enter the student's data, present the current data, read from and save the data to a file, or exit the program. The assignment builds on the concepts learned in Assignment 05, incorporating classes and functions to read from a file and write to a file as well as process the inputs from the user. It also implemented error handling on reading input files and checks for unallowed characters for students' first name, last name, and course. This document outlines the steps taken to complete the assignment.

Preparation

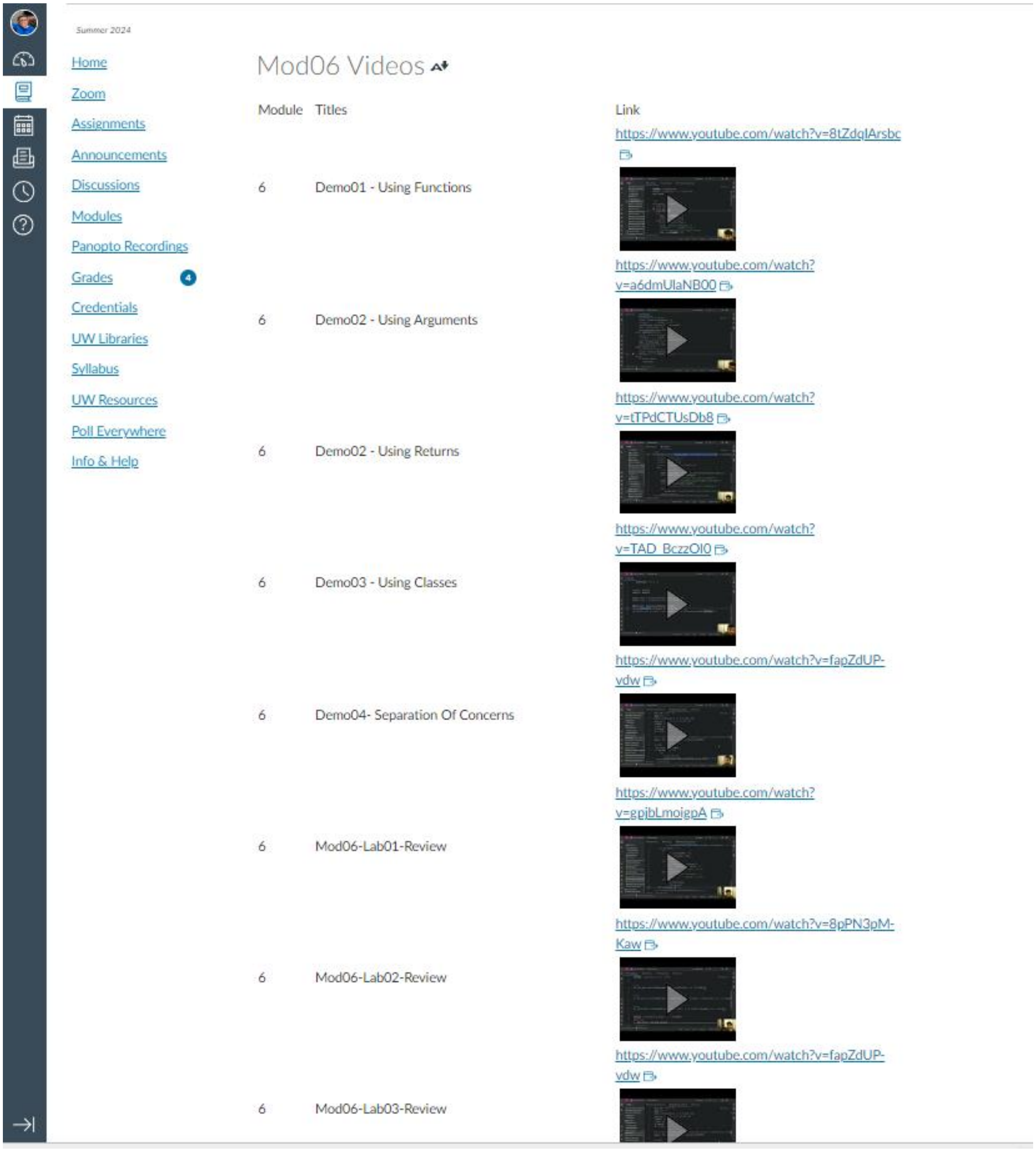
To prepare for this assignment, I read the Module 06 notes on classes and functions. The notes extended the module 05 learnings on working with files, as well as how to feed data from files into the code and vice versa, but now utilizing functions and classes to perform the tasks. The notes samples on working with classes and functions were particularly useful in preparation for the coding assignment.

Writing and executing Module 06 Python script

After finishing reading module 06, I watched the module 06 YouTube videos from Professor Arya Ref [1] (see Figure 1), and having tried out the example Python code in the folders "Demos" and "LabAnswers," (see Figure 2), I felt prepared to start the programming assignment.

I used the provided Assignment06-Starter.py file, but I soon realized that most of the structure of the code had to be changed to be used in the two classes and the multiple functions required for the assignment. The readings and videos made writing the script (Figure 3) much easier, especially the Modul06-Lab03. The steps I followed can be described as:

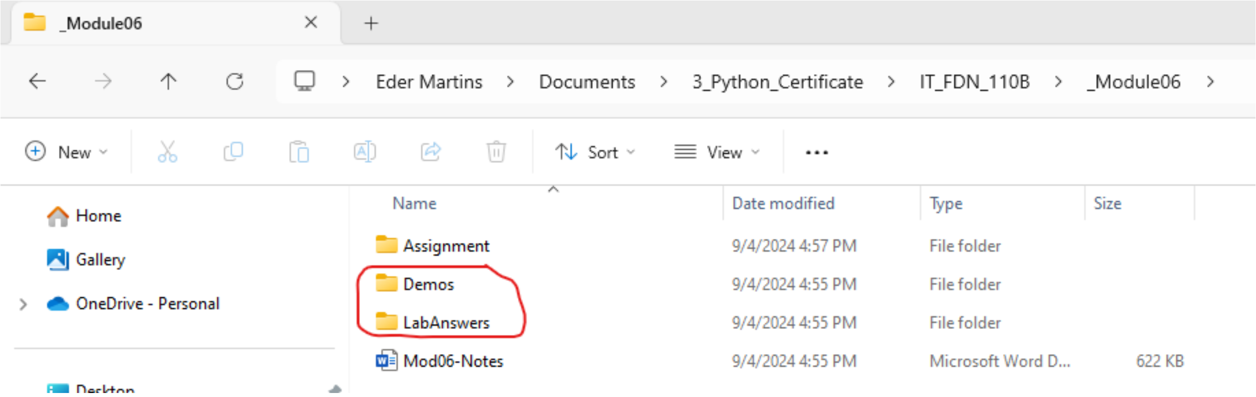
1. *Update the header*: with my name and current date
2. *Define constants*: set the constant for the menu options (‘MENU’) and the file name (‘FILE_NAME’) where the data will be stored and/or read from.
3. *Define variables*: initialized variables for storing user input such as menu_choice, students (list of student dictionaries), student_first_name, student_last_name, and course_name..
4. *Check for existing data*: Implemented file handling by reading from the file (Enrollments.json) using the FileProcessor.read_data_from_file() method. If the file exists, the data is loaded into the students list. Handled errors if the file doesn't exist.
5. *Display the menu:* Use a while True loop to continuously display the menu and prompt the user to select an option (register a student, show current data, save data to a file, or exit the program).
6. *Handle user choices*: Based on the user's selection, perform the following:
    - Register a student: Prompts the user to enter student details (first name, last name, and course name), which are appended to the students list. Error handling is added to ensure that only valid alphabetic names are entered, raising ValueError for invalid input.
    - Show current data: Displays all registered students and their enrolled courses.
    - Save data to a file: Saves the current student data to the Enrollments.json file using FileProcessor.write_data_to_file() method.
7. *Test the Program*: run the program in PyCharm (Figure 4) and from the console (Figure 5) to ensure it worked as expected. Verified that the output was saved correctly in the Enrollments.json file (Figure 6).



Mod06 Videos

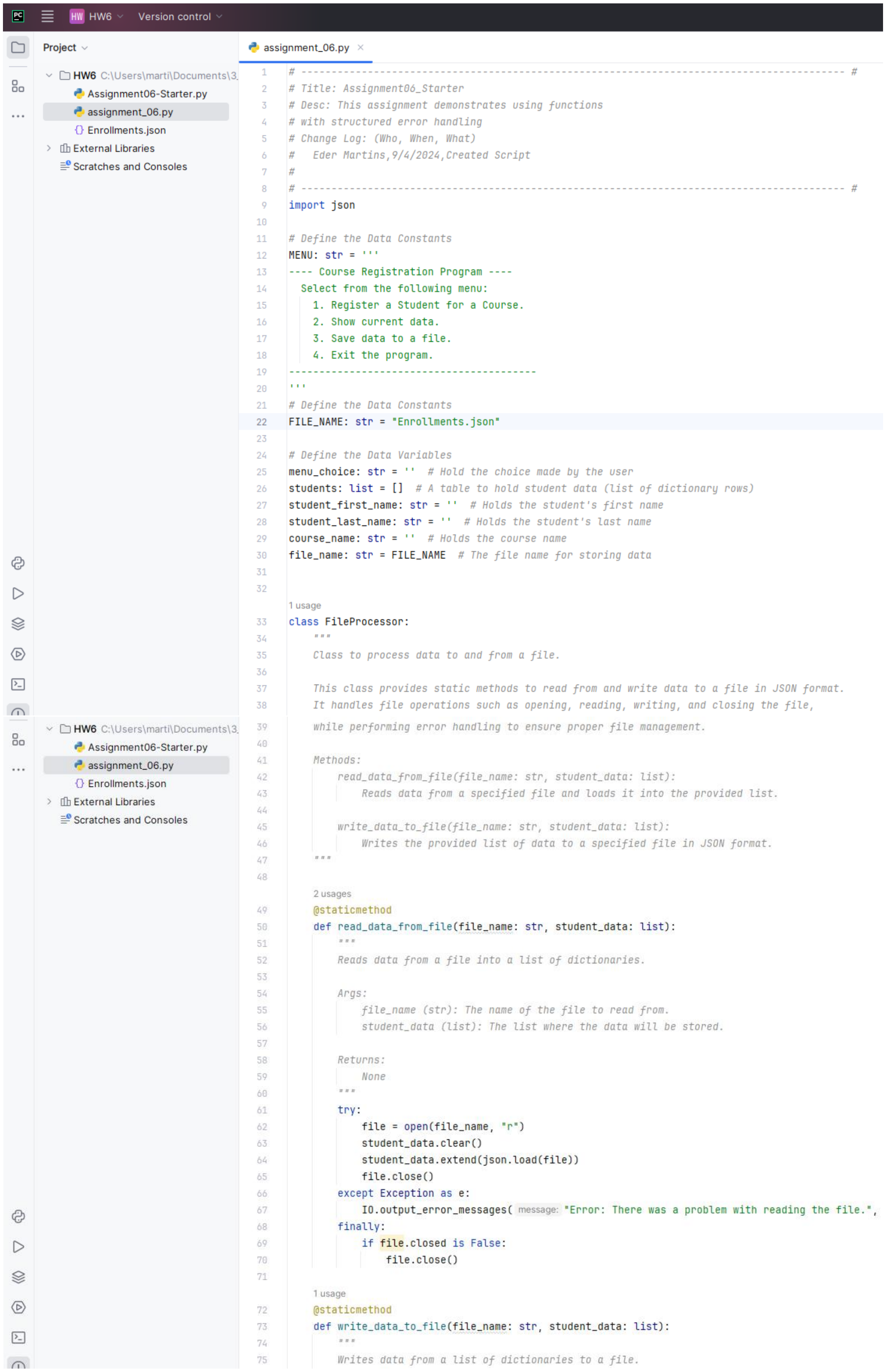
Module	Titles	Link
6	Demo01 - Using Functions	https://www.youtube.com/watch?v=8tZdqlArsbc
6	Demo02 - Using Arguments	https://www.youtube.com/watch?v=a6dmUlaNB00
6	Demo02 - Using Returns	https://www.youtube.com/watch?v=tTPdCTUsDb8
6	Demo03 - Using Classes	https://www.youtube.com/watch?v=TAD_BczzOI0
6	Demo04- Separation Of Concerns	https://www.youtube.com/watch?v=fapZdUP-vdw
6	Mod06-Lab01-Review	https://www.youtube.com/watch?v=gpjbLmoigpA
6	Mod06-Lab02-Review	https://www.youtube.com/watch?v=8pPN3pM-Kaw
6	Mod06-Lab03-Review	https://www.youtube.com/watch?v=fapZdUP-vdw

Figure 1 Module 06 Videos (screenshot from video's list in Canvas)


_Module06
Eder Martins > Documents > 3_Python_Certificate > IT_FDN_110B > _Module06 >
New
Sort
View
Home
Gallery
OneDrive - Personal
Desktop

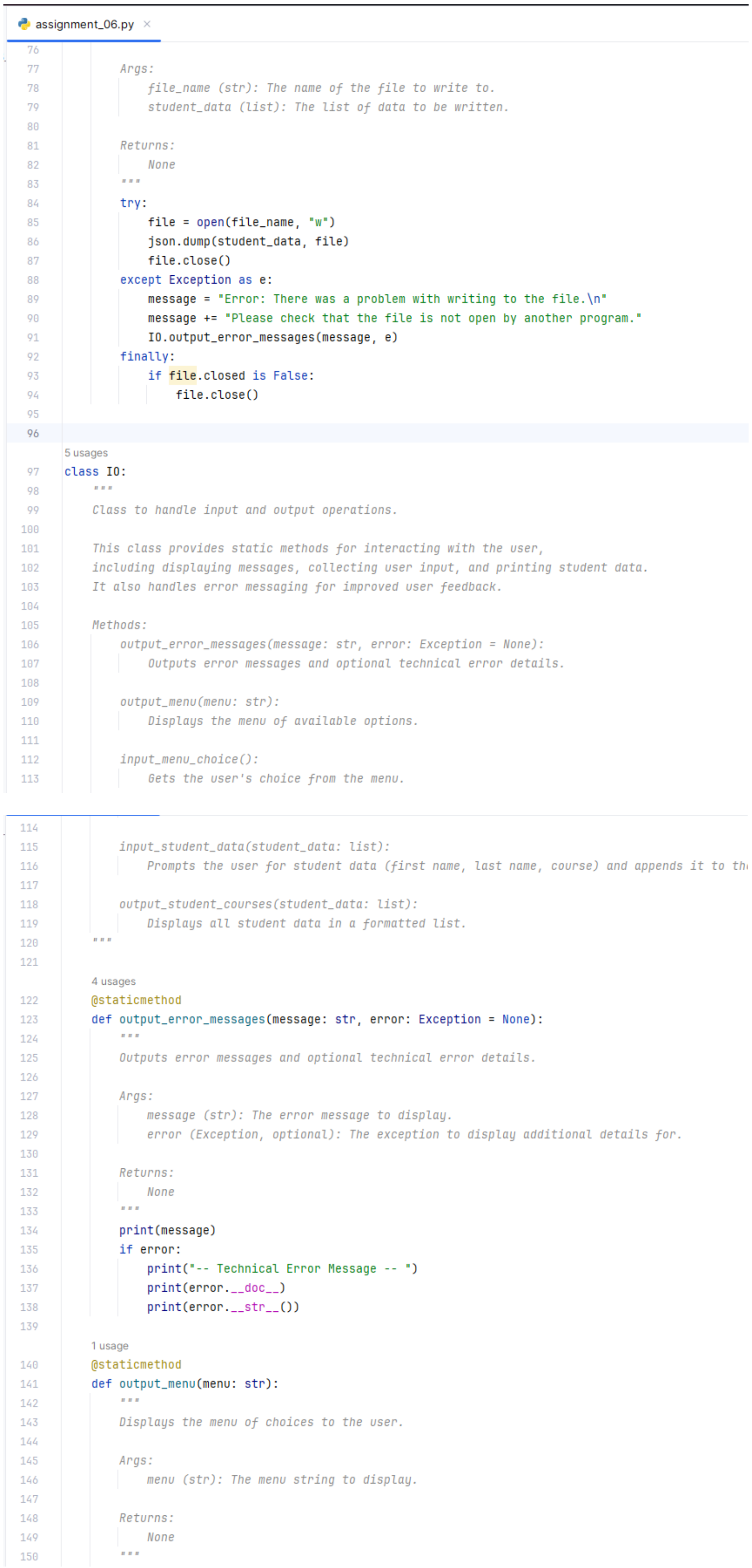
Name	Date modified	Type	Size
Assignment	9/4/2024 4:57 PM	File folder	
Demos	9/4/2024 4:55 PM	File folder	
LabAnswers	9/4/2024 4:55 PM	File folder	
Mod06-Notes	9/4/2024 4:55 PM	Microsoft Word D...	622 KB

Figure 2 Module 06 subfolders “Demos” and “LabAnswers” highlighted



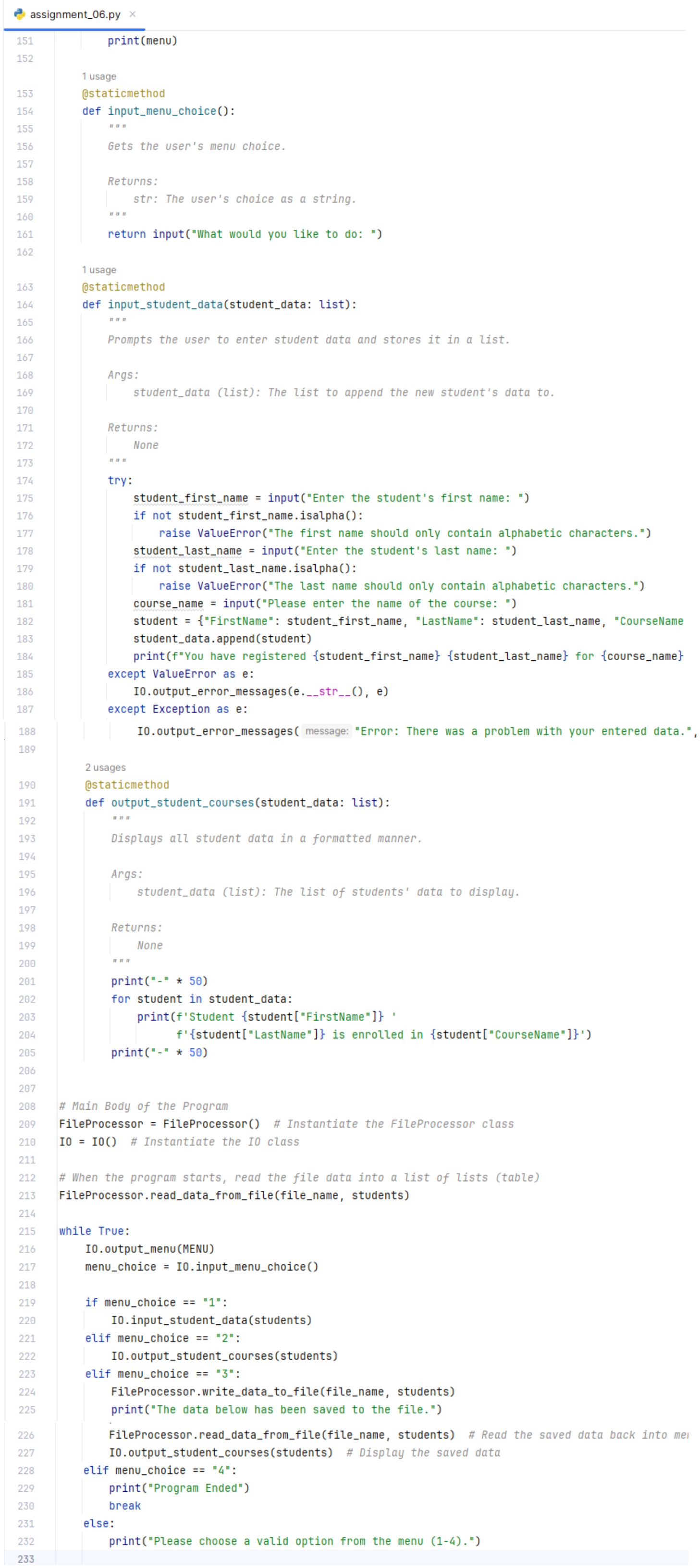
```python
# ------------------------------------------------------------------------------------------ #
# Title: Assignment06_Starter
# Desc: This assignment demonstrates using functions
# with structured error handling
# Change Log: (Who, When, What)
#   Eder Martins,9/4/2024,Created Script
#
# ------------------------------------------------------------------------------------------ #
import json

# Define the Data Constants
MENU: str = '''
---- 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.
-----------------------------------------
'''
# Define the Data Constants
FILE_NAME: str = "Enrollments.json"

# Define the Data Variables
menu_choice: str = ''  # Hold the choice made by the user
students: list = []  # A table to hold student data (list of dictionary rows)
student_first_name: str = ''  # Holds the student's first name
student_last_name: str = ''  # Holds the student's last name
course_name: str = ''  # Holds the course name
file_name: str = FILE_NAME  # The file name for storing data


class FileProcessor:
    """
    Class to process data to and from a file.

    This class provides static methods to read from and write data to a file in JSON format.
    It handles file operations such as opening, reading, writing, and closing the file,
    while performing error handling to ensure proper file management.

    Methods:
        read_data_from_file(file_name: str, student_data: list):
            Reads data from a specified file and loads it into the provided list.

        write_data_to_file(file_name: str, student_data: list):
            Writes the provided list of data to a specified file in JSON format.
    """

    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        """
        Reads data from a file into a list of dictionaries.

        Args:
            file_name (str): The name of the file to read from.
            student_data (list): The list where the data will be stored.

        Returns:
            None
        """
        try:
            file = open(file_name, "r")
            student_data.clear()
            student_data.extend(json.load(file))
            file.close()
        except Exception as e:
            IO.output_error_messages( "Error: There was a problem with reading the file.",
        finally:
            if file.closed is False:
                file.close()

    @staticmethod
    def write_data_to_file(file_name: str, student_data: list):
        """
        Writes data from a list of dictionaries to a file.
```

Figure 3 Python Script in PyCharm using Classes and Functions



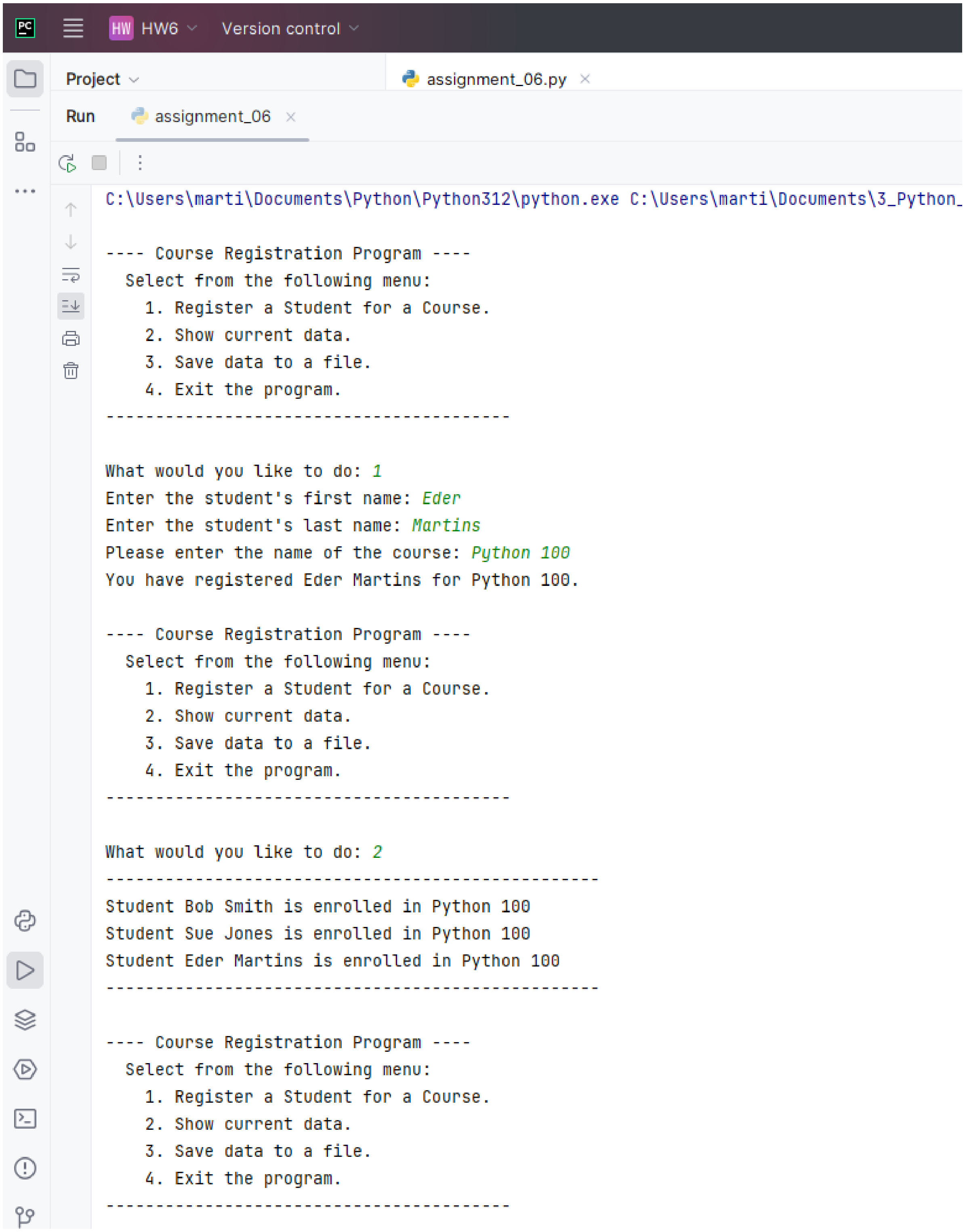
```python

        Args:
            file_name (str): The name of the file to write to.
            student_data (list): The list of data to be written.

        Returns:
            None
        """
        try:
            file = open(file_name, "w")
            json.dump(student_data, file)
            file.close()
        except Exception as e:
            message = "Error: There was a problem with writing to the file.\n"
            message += "Please check that the file is not open by another program."
            IO.output_error_messages(message, e)
        finally:
            if file.closed is False:
                file.close()


class IO:
    """
    Class to handle input and output operations.

    This class provides static methods for interacting with the user,
    including displaying messages, collecting user input, and printing student data.
    It also handles error messaging for improved user feedback.

    Methods:
        output_error_messages(message: str, error: Exception = None):
            Outputs error messages and optional technical error details.

        output_menu(menu: str):
            Displays the menu of available options.

        input_menu_choice():
            Gets the user's choice from the menu.

        input_student_data(student_data: list):
            Prompts the user for student data (first name, last name, course) and appends it to th

        output_student_courses(student_data: list):
            Displays all student data in a formatted list.
    """

    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        """
        Outputs error messages and optional technical error details.

        Args:
            message (str): The error message to display.
            error (Exception, optional): The exception to display additional details for.

        Returns:
            None
        """
        print(message)
        if error:
            print("-- Technical Error Message -- ")
            print(error.__doc__)
            print(error.__str__())

    @staticmethod
    def output_menu(menu: str):
        """
        Displays the menu of choices to the user.

        Args:
            menu (str): The menu string to display.

        Returns:
            None
        """
```

Figure 3 Python Script in PyCharm using Classes and Functions (contnd)



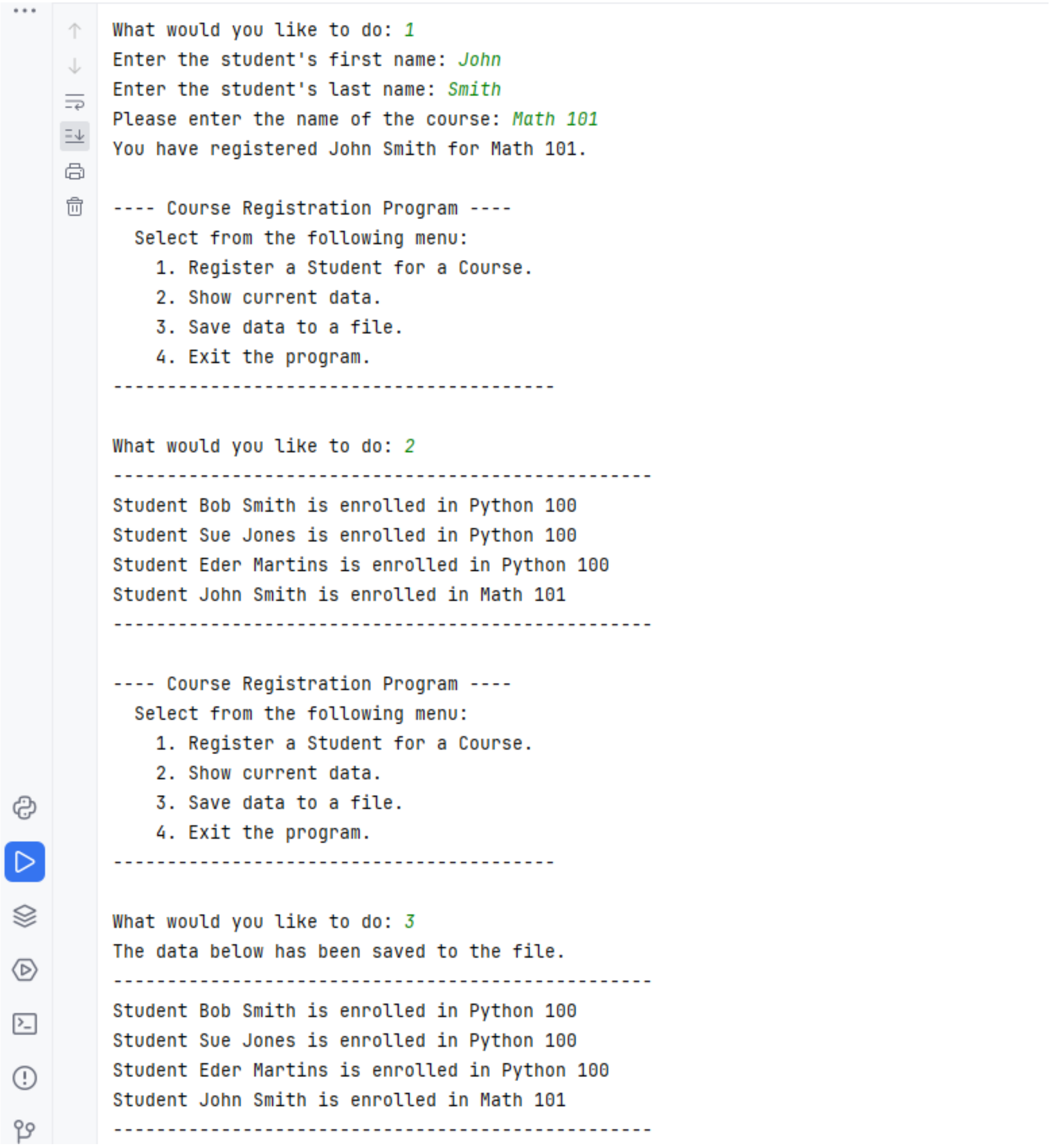
```python
        print(menu)

    1 usage
    @staticmethod
    def input_menu_choice():
        """
        Gets the user's menu choice.

        Returns:
            str: The user's choice as a string.
        """
        return input("What would you like to do: ")

    1 usage
    @staticmethod
    def input_student_data(student_data: list):
        """
        Prompts the user to enter student data and stores it in a list.

        Args:
            student_data (list): The list to append the new student's data to.

        Returns:
            None
        """
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The first name should only contain alphabetic characters.")
            student_last_name = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The last name should only contain alphabetic characters.")
            course_name = input("Please enter the name of the course: ")
            student = {"FirstName": student_first_name, "LastName": student_last_name, "CourseName
            student_data.append(student)
            print(f"You have registered {student_first_name} {student_last_name} for {course_name}
        except ValueError as e:
            IO.output_error_messages(e.__str__(), e)
        except Exception as e:
            IO.output_error_messages( message: "Error: There was a problem with your entered data.",

    2 usages
    @staticmethod
    def output_student_courses(student_data: list):
        """
        Displays all student data in a formatted manner.

        Args:
            student_data (list): The list of students' data to display.

        Returns:
            None
        """
        print("-" * 50)
        for student in student_data:
            print(f'Student {student["FirstName"]} '
                  f'{student["LastName"]} is enrolled in {student["CourseName"]}')
        print("-" * 50)


# Main Body of the Program
FileProcessor = FileProcessor()  # Instantiate the FileProcessor class
IO = IO()  # Instantiate the IO class

# When the program starts, read the file data into a list of lists (table)
FileProcessor.read_data_from_file(file_name, students)

while True:
    IO.output_menu(MENU)
    menu_choice = IO.input_menu_choice()

    if menu_choice == "1":
        IO.input_student_data(students)
    elif menu_choice == "2":
        IO.output_student_courses(students)
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(file_name, students)
        print("The data below has been saved to the file.")
        FileProcessor.read_data_from_file(file_name, students)  # Read the saved data back into mer
        IO.output_student_courses(students)  # Display the saved data
    elif menu_choice == "4":
        print("Program Ended")
        break
    else:
        print("Please choose a valid option from the menu (1-4).")
```

Figure 3 Python Script in PyCharm using Classes and Functions (contnd)

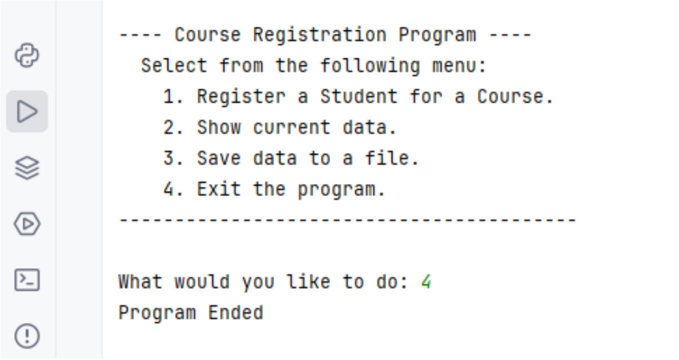


```
C:\Users\marti\Documents\Python\Python312\python.exe C:\Users\marti\Documents\3_Python_

---- 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: Eder
Enter the student's last name: Martins
Please enter the name of the course: Python 100
You have registered Eder Martins 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 Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Eder Martins 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.
-----------------------------------------
```

Figure 4 Executing Python Script in PyCharm

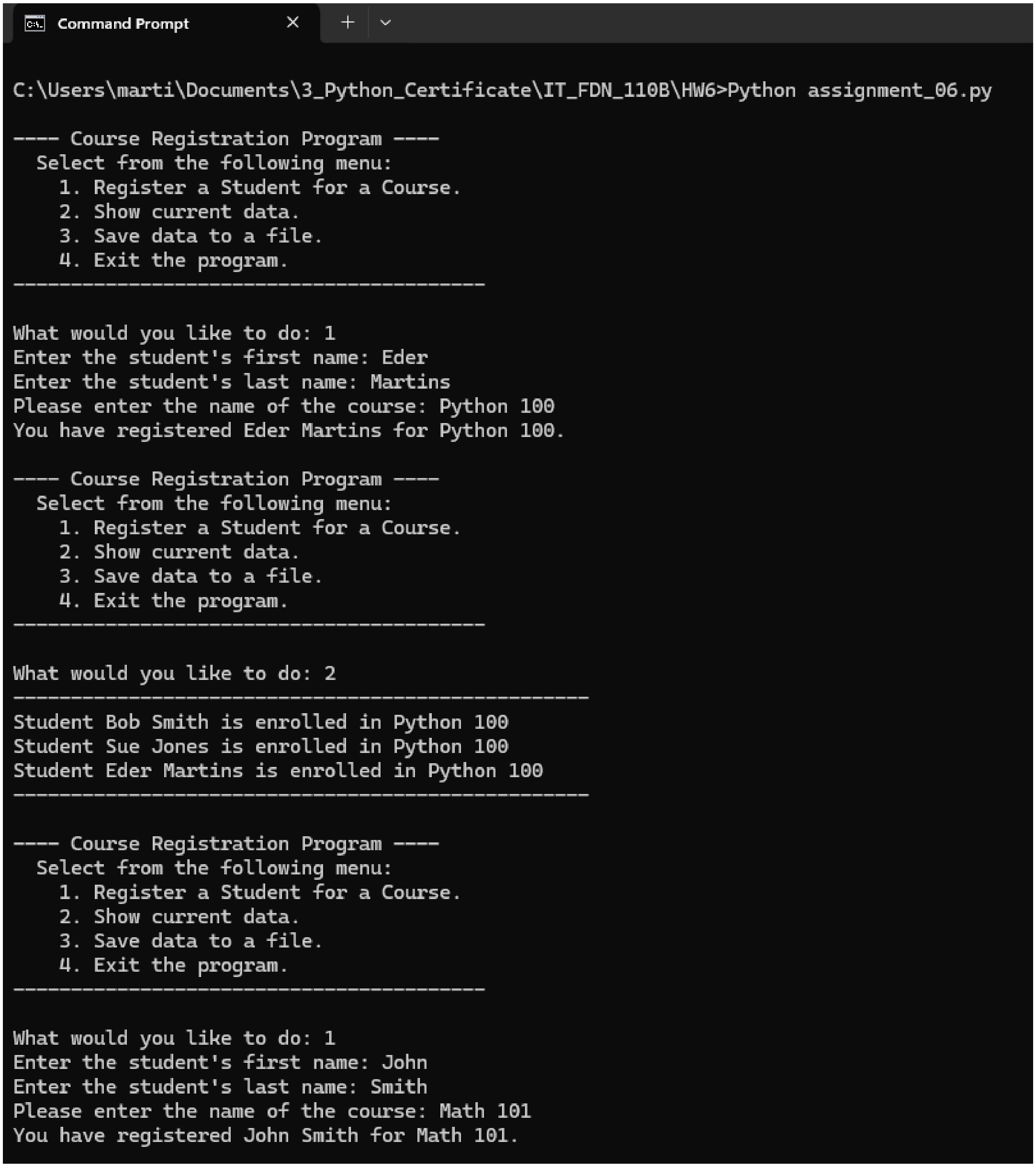


```
What would you like to do: 1
Enter the student's first name: John
Enter the student's last name: Smith
Please enter the name of the course: Math 101
You have registered John Smith for Math 101.

---- 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 Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Eder Martins is enrolled in Python 100
Student John Smith is enrolled in Math 101
--------------------------------------------------

---- 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 data below has been saved to the file.
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Eder Martins is enrolled in Python 100
Student John Smith is enrolled in Math 101
--------------------------------------------------
```



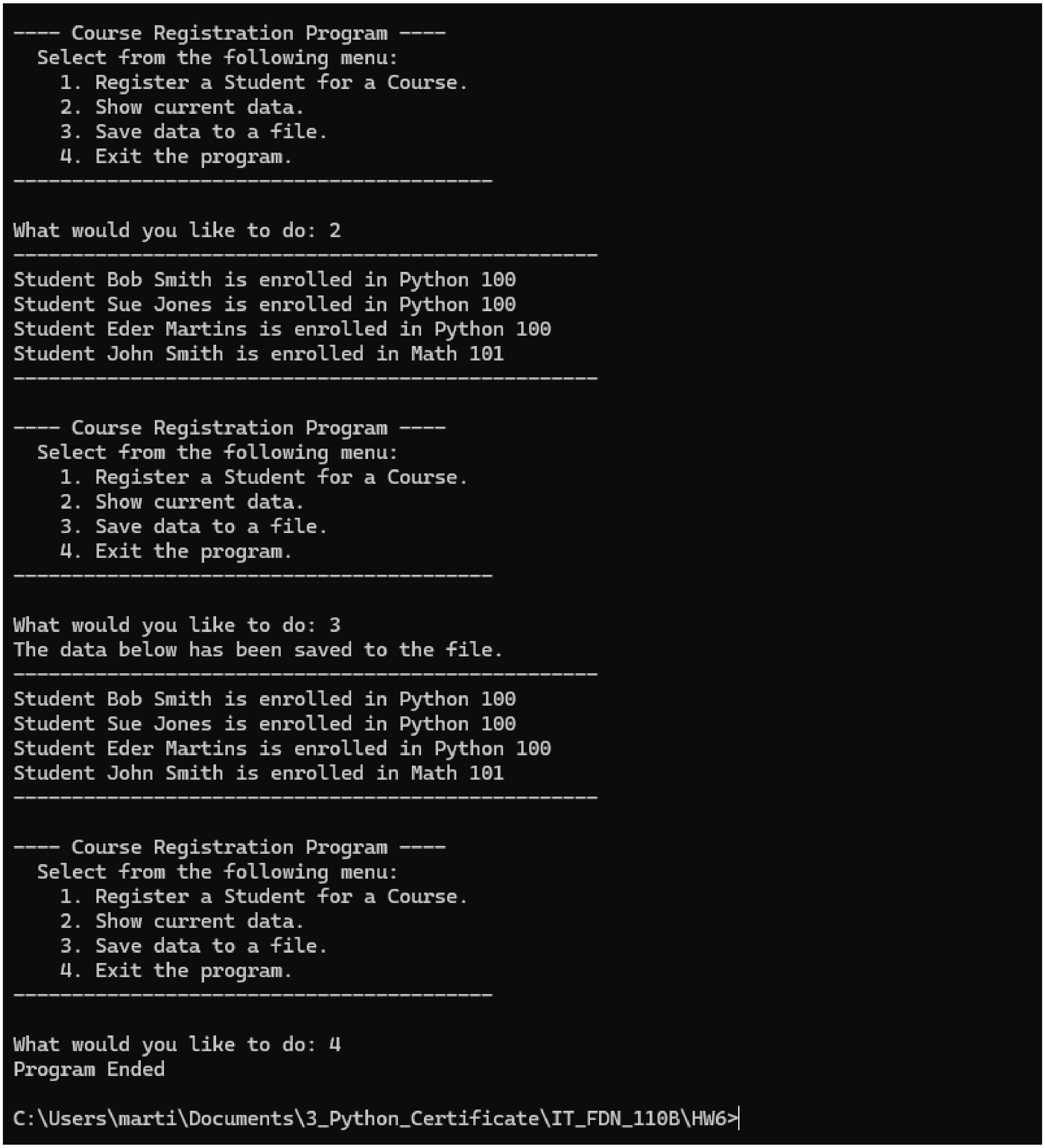
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------

What would you like to do: 4
Program Ended
```

Figure 4 Executing Python Script in PyCharm (contnd)



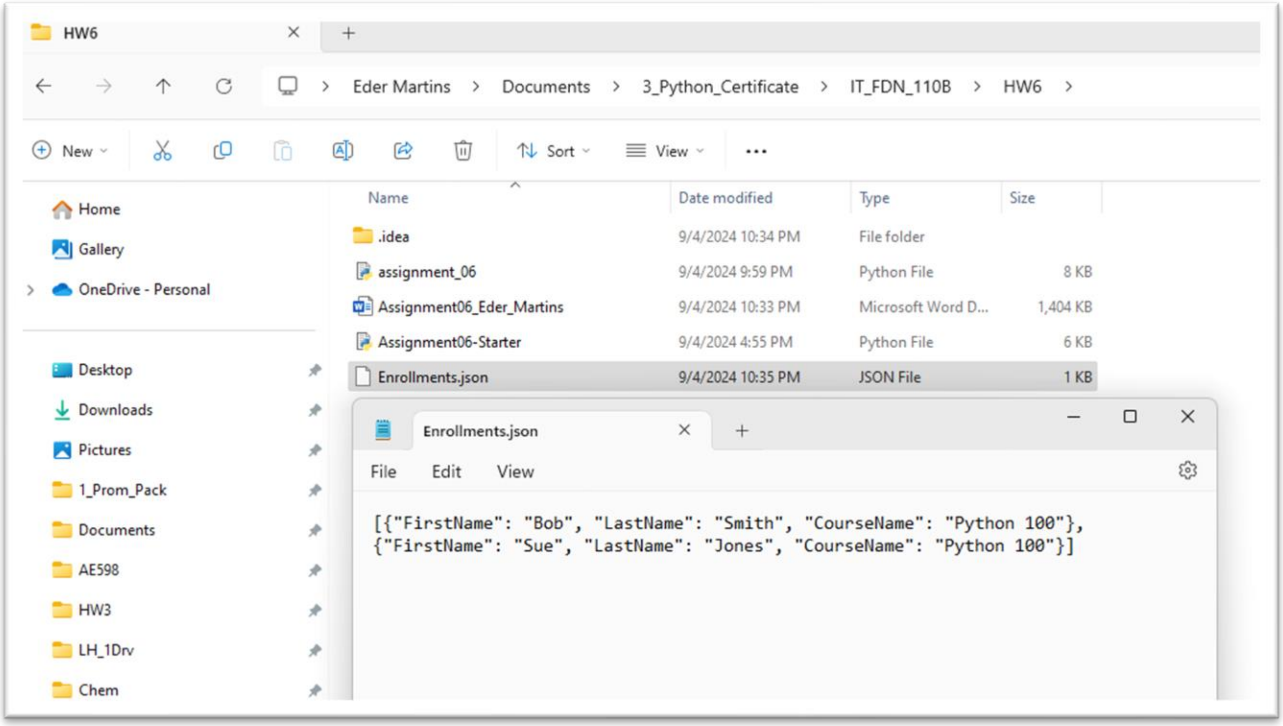
```
C:\Users\marti\Documents\3_Python_Certificate\IT_FDN_110B\HW6>Python assignment_06.py

---- 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: 1
Enter the student's first name: Eder
Enter the student's last name: Martins
Please enter the name of the course: Python 100
You have registered Eder Martins 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 Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Eder Martins 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: 1
Enter the student's first name: John
Enter the student's last name: Smith
Please enter the name of the course: Math 101
You have registered John Smith for Math 101.
```

Figure 5 Executing Python Script in the Command Prompt



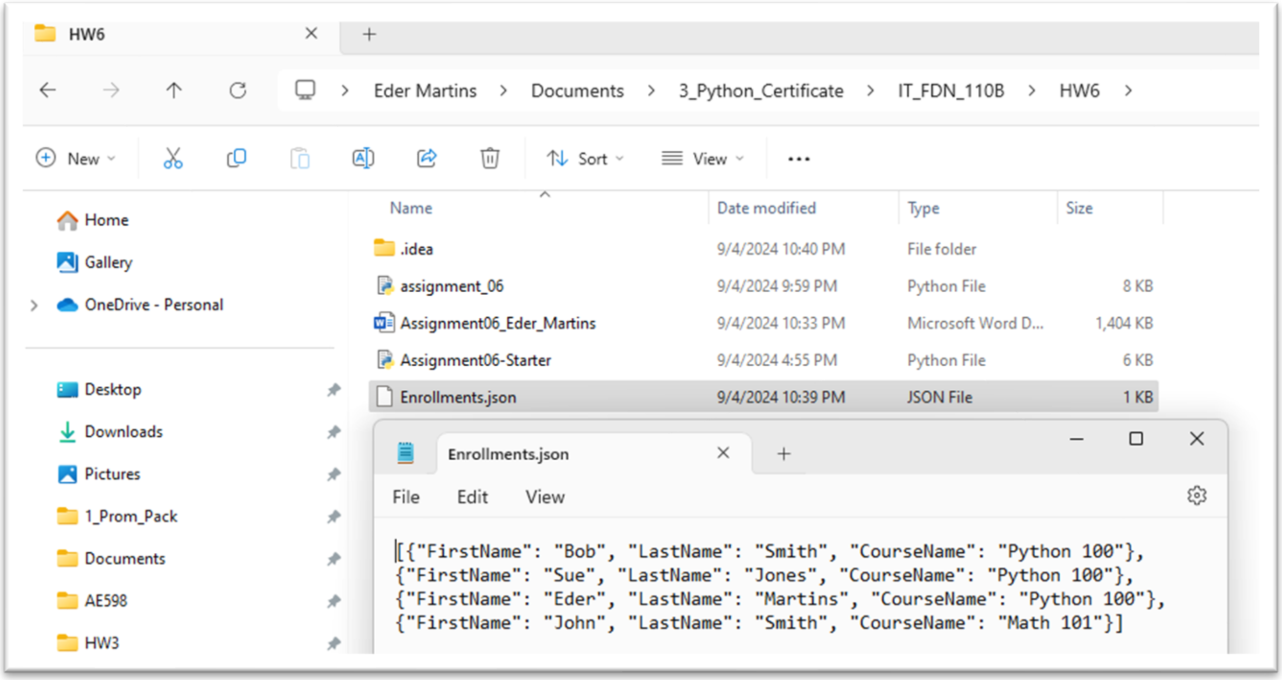
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------

What would you like to do: 2
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Eder Martins is enrolled in Python 100
Student John Smith is enrolled in Math 101
--------------------------------------------------

---- 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 data below has been saved to the file.
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Eder Martins is enrolled in Python 100
Student John Smith is enrolled in Math 101
--------------------------------------------------

---- 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

C:\Users\marti\Documents\3_Python_Certificate\IT_FDN_110B\HW6>
```

Figure 5 Executing Python Script in the Command Prompt (contnd)



Name	Date modified	Type	Size
.idea	9/4/2024 10:34 PM	File folder	
assignment_06	9/4/2024 9:59 PM	Python File	8 KB
Assignment06_Eder_Martins	9/4/2024 10:33 PM	Microsoft Word D...	1,404 KB
Assignment06-Starter	9/4/2024 4:55 PM	Python File	6 KB
Enrollments.json	9/4/2024 10:35 PM	JSON File	1 KB

```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"},
{"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"}]
```



Name	Date modified	Type	Size
.idea	9/4/2024 10:40 PM	File folder	
assignment_06	9/4/2024 9:59 PM	Python File	8 KB
Assignment06_Eder_Martins	9/4/2024 10:33 PM	Microsoft Word D...	1,404 KB
Assignment06-Starter	9/4/2024 4:55 PM	Python File	6 KB
Enrollments.json	9/4/2024 10:39 PM	JSON File	1 KB

```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"},
{"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"},
{"FirstName": "Eder", "LastName": "Martins", "CourseName": "Python 100"},
{"FirstName": "John", "LastName": "Smith", "CourseName": "Math 101"}]
```

Figure 6 Enrollments.json file content before and after running the script

Summary

To complete Assignment 06, I followed a structured approach that integrated new and previously learned concepts. The primary focus was on leveraging functions, classes, and the separation of concerns pattern to improve the organization and efficiency of the program. I learned how to define and use constants, variables, functions, and classes to manage data collection and processing. Key new concepts included organizing code with functions and classes, validating user input, and implementing structured error handling to ensure the program operated smoothly. Additionally, I applied earlier concepts such as conditional logic, loops, and comparison operators to build a functional program. Through careful reading of the module materials, watching tutorial videos, and experimenting with code samples, I developed the skills necessary to create a Python program that met all the assignment's requirements. The final program displayed a menu with options to register students, show current data, save data to a file, and exit the program. It ensured data integrity by validating user inputs and utilized dictionaries to store student records. The program also effectively managed data persistence by saving and retrieving records using JSON files. By successfully integrating these programming techniques, I was able to create a user-friendly course registration system that demonstrates how to organize, validate, and store data efficiently. This assignment enhanced my understanding of how to use functions and classes in Python to write cleaner, more modular code.

GitHub Repository: https://github.com/martins-eder/IntroToProg-Python-Mod06

Reference

[1] Arya, Anubhaw, *Module 06 Videos*. Available at YouTube: www.youtube.com/@arya0-uw