Name: William McGrath

Date: August 9, 2025

Class: IT FDN 110 A; Term: Summer 2025

Assignment: Assignment 05

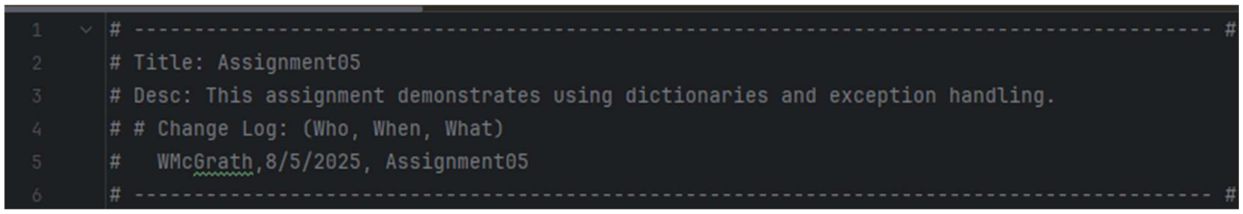
Module 05 - Advanced Collections and Error Handling

Introduction

This document describes the steps I took to complete ‘Task 4: Create a program’ of ‘Assignment 05’. The purpose of the program is to help the user enter student registration information into a database. The program has four (4) menu options. Three (3) options perform tasks, and one (1) option exists from the program. A while loop is used to allow the user to continue to perform the tasks of the first three options.

Script header

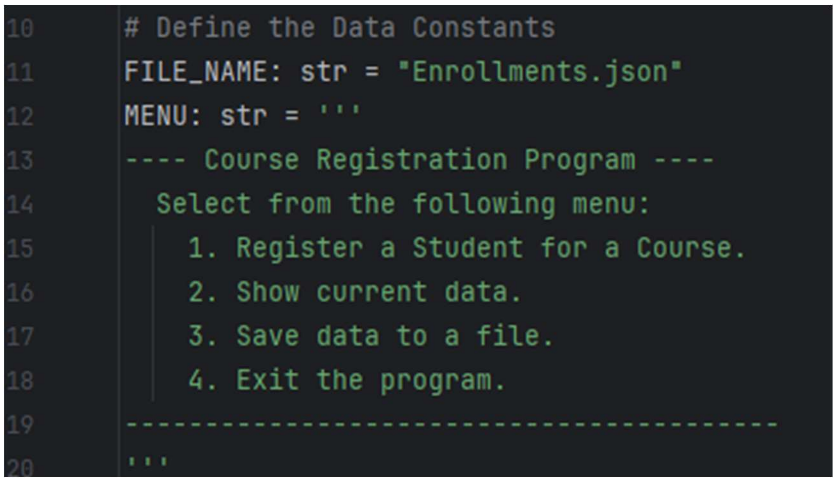
A file header was added to capture information about the file.



```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries and exception handling.
# # Change Log: (Who, When, What)
#   WMcGrath,8/5/2025, Assignment05
# ------------------------------------------------------------------------------------------ #
```

Define data constants

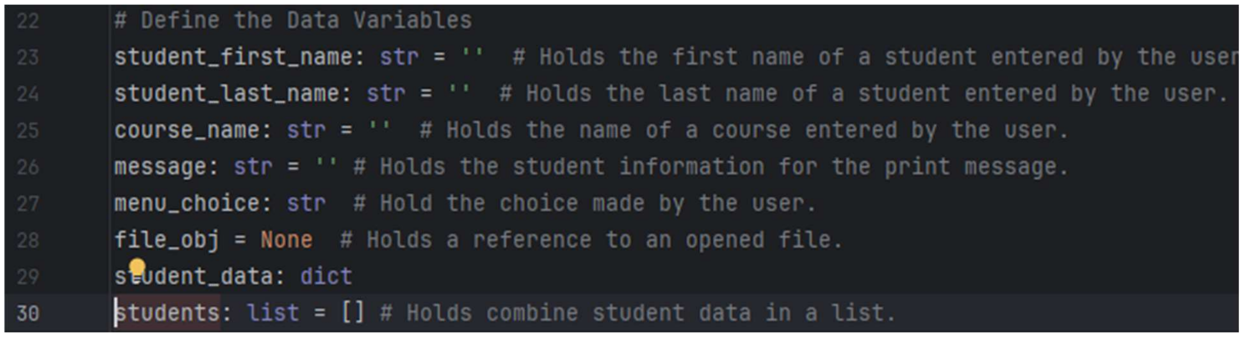
Next, I defined two constant variables. The first constant defines the name of the JSON file and the second sets forth menu options for the user.



```
# Define the Data Constants
FILE_NAME: str = "Enrollments.json"
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 data variables

The following data variables were added to store user inputs.



```
# Define the Data Variables
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.
message: str = '' # Holds the student information for the print message.
menu_choice: str  # Hold the choice made by the user.
file_obj = None  # Holds a reference to an opened file.
student_data: dict
students: list = [] # Holds combine student data in a list.
```

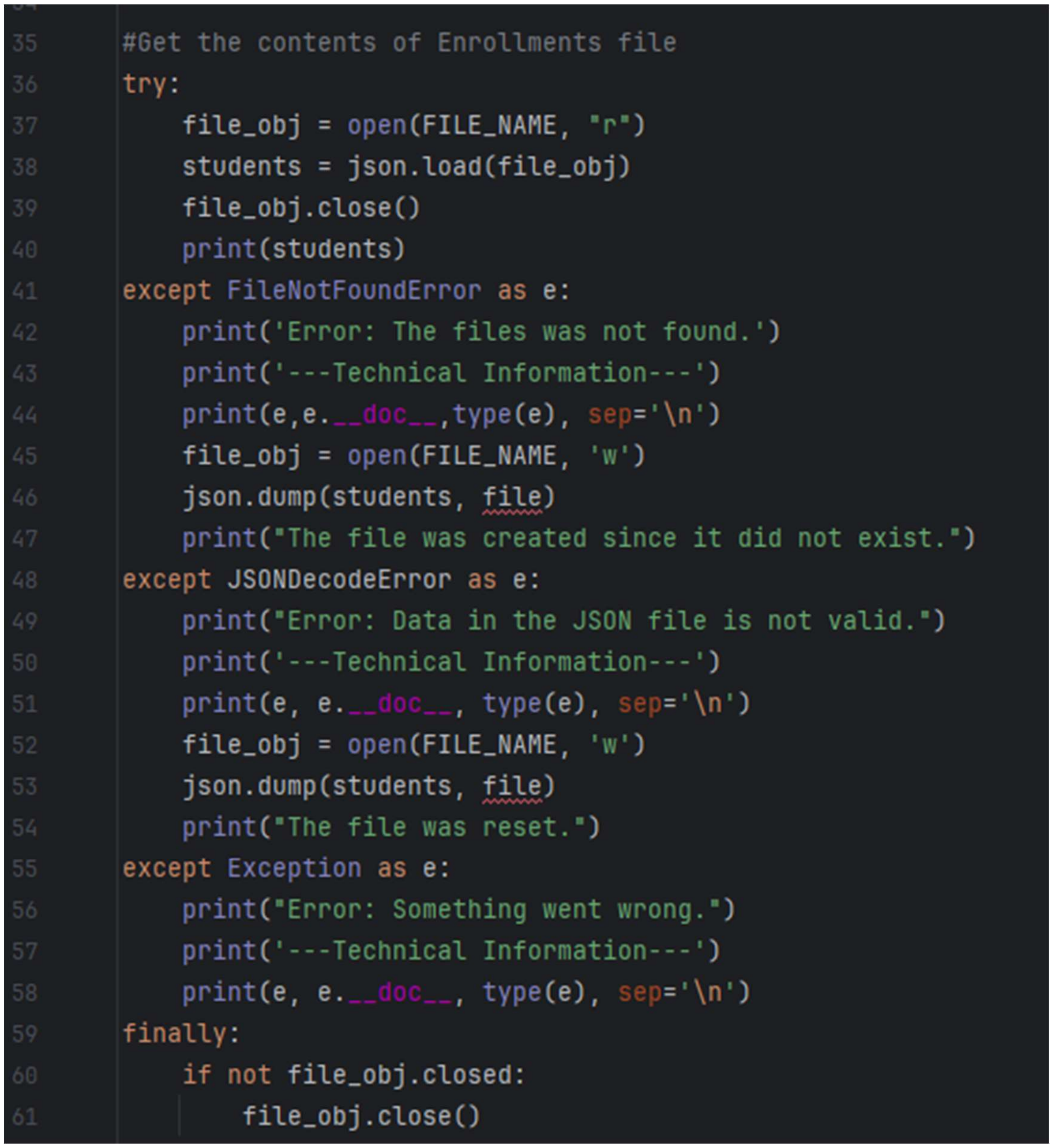
The ‘student_data’ variable is a dictionary set to an empty dictionary. The ‘students’ variable is a list set to an empty list.

Present and process the data

Read the contents of Enrollments.json

The program starts by assigning the contents Enrollments.json to the students list.

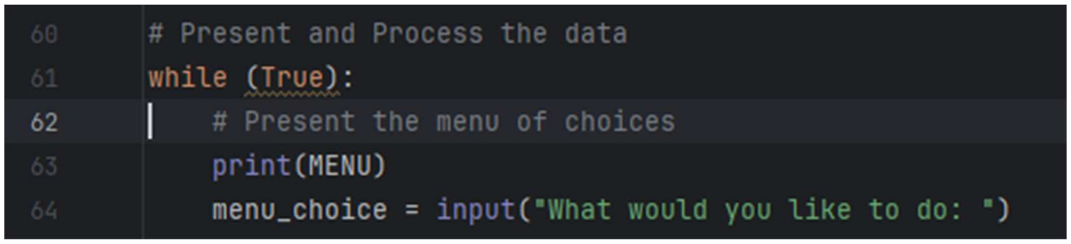
If an error is encountered while opening the json file, a try-except construct was used to tell the user to check if the file exists. The try-except construct also ensures that the json file is closed.



```python
#Get the contents of Enrollments file
try:
    file_obj = open(FILE_NAME, "r")
    students = json.load(file_obj)
    file_obj.close()
    print(students)
except FileNotFoundError as e:
    print('Error: The files was not found.')
    print('---Technical Information---')
    print(e,e.__doc__,type(e), sep='\n')
    file_obj = open(FILE_NAME, 'w')
    json.dump(students, file)
    print("The file was created since it did not exist.")
except JSONDecodeError as e:
    print("Error: Data in the JSON file is not valid.")
    print('---Technical Information---')
    print(e, e.__doc__, type(e), sep='\n')
    file_obj = open(FILE_NAME, 'w')
    json.dump(students, file)
    print("The file was reset.")
except Exception as e:
    print("Error: Something went wrong.")
    print('---Technical Information---')
    print(e, e.__doc__, type(e), sep='\n')
finally:
    if not file_obj.closed:
        file_obj.close()
```

Display menu

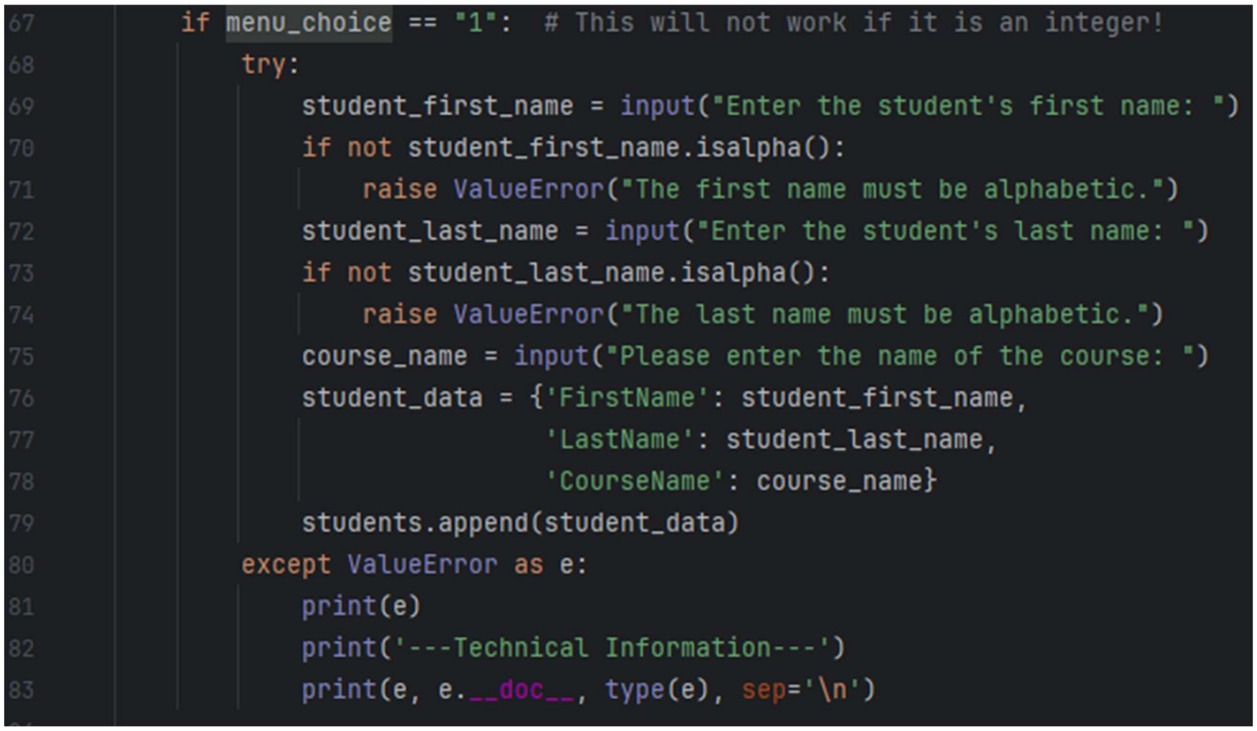
The program uses a while loop to allow the user to access the various functions of the program. Each loop starts by printing the MENU variable, which is a menu of options. There are four (4) options on the menu.



```
# Present and Process the data
while (True):
    # Present the menu of choices
    print(MENU)
    menu_choice = input("What would you like to do: ")
```

Option 1: Register a student for a course.

When menu choice 1 is selected, the user is asked to enter student class registration information. Student information is assigned to the 'student_data' dictionary. 'student_data' is then appended to the 'students' list. A try-except construct is used to help ensure valid information is entered.

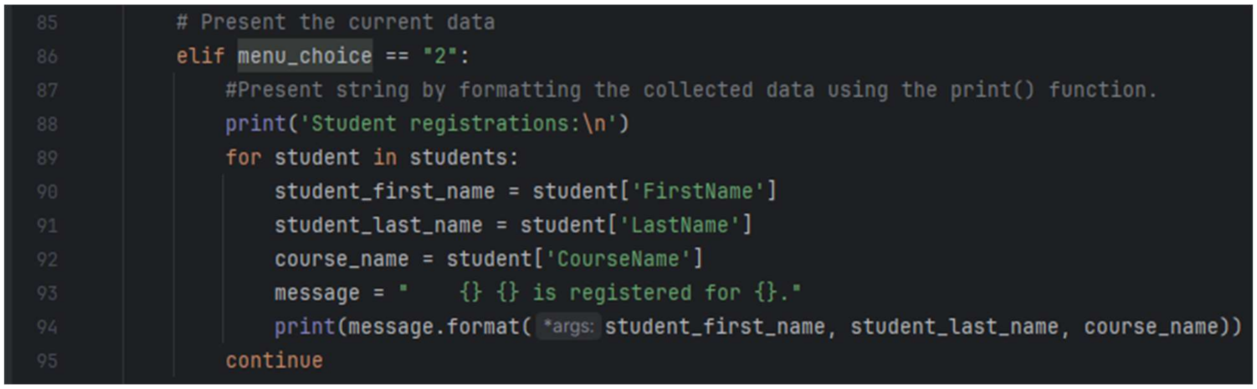


```
    if menu_choice == "1":  # This will not work if it is an integer!
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The first name must be alphabetic.")
            student_last_name = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The last name must be alphabetic.")
            course_name = input("Please enter the name of the course: ")
            student_data = {'FirstName': student_first_name,
                            'LastName': student_last_name,
                            'CourseName': course_name}
            students.append(student_data)
        except ValueError as e:
            print(e)
            print('---Technical Information---')
            print(e, e.__doc__, type(e), sep='\n')
```

The user can select option 1 multiple times to enter new registrations. If option 1 is not chosen, then an *elif clause* is used to again evaluate the user's menu choice to see if option 2 was chosen.

Option 2: Show current data.

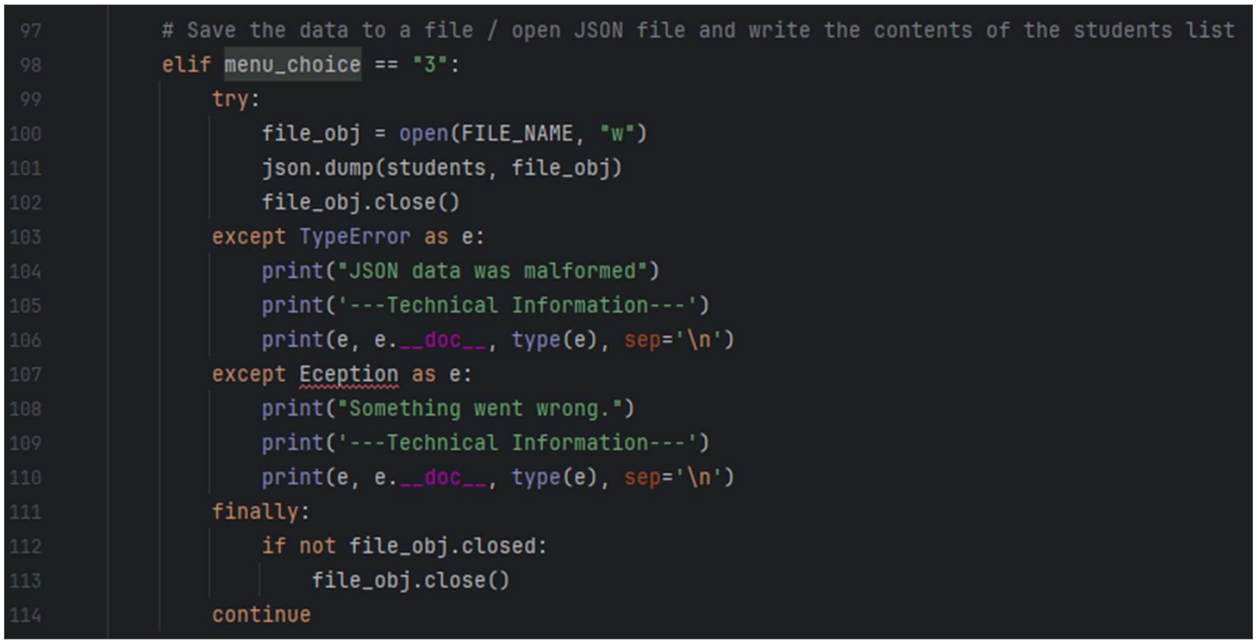
If the user chooses "Option 2: Show current data", then a for loop displays a message with each student's information collected in the 'students' variable. If option 2 is not chosen, then an *elif clause* is used to evaluate the user's menu choice to see if option 3 was chosen.



```
    # Present the current data
    elif menu_choice == "2":
        #Present string by formatting the collected data using the print() function.
        print('Student registrations:\n')
        for student in students:
            student_first_name = student['FirstName']
            student_last_name = student['LastName']
            course_name = student['CourseName']
            message = "    {} {} is registered for {}."
            print(message.format( *args: student_first_name, student_last_name, course_name))
        continue
```

Option 3: Save the data to a file.

If the user chooses "Option 3: Save data to a file", then the program saves the contents of the 'students' list a csv file containing the user's inputs. A for loop writes each student's name and course name to a json file. A try-except construct helps provide the user more information should they encounter an error in while writing to Enrollments. json.

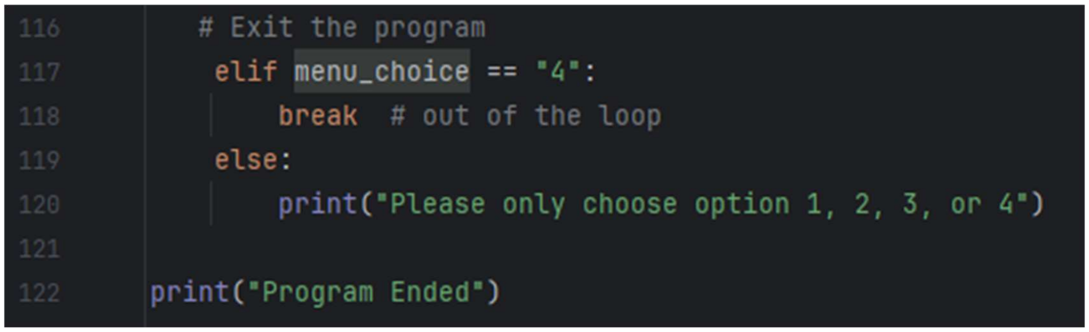


```
    # Save the data to a file / open JSON file and write the contents of the students list
    elif menu_choice == "3":
        try:
            file_obj = open(FILE_NAME, "w")
            json.dump(students, file_obj)
            file_obj.close()
        except TypeError as e:
            print("JSON data was malformed")
            print('---Technical Information---')
            print(e, e.__doc__, type(e), sep='\n')
        except Eception as e:
            print("Something went wrong.")
            print('---Technical Information---')
            print(e, e.__doc__, type(e), sep='\n')
        finally:
            if not file_obj.closed:
                file_obj.close()
        continue
```

If option 3 is not chosen, then an *elif clause* is used to again evaluate the user's menu choice to see if option 4 was chosen.

Option 4: Exit the program

If the user chooses "Option 4: Exit the program", the elif includes the break statement to end the loop. After a while loop, a print() function confirms to the user that the program has ended.



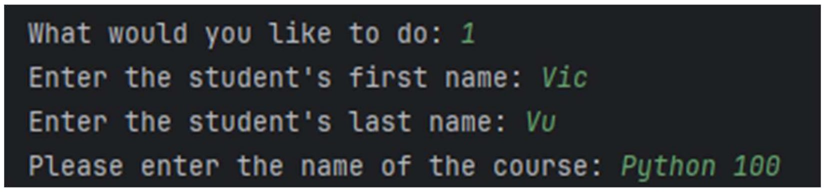
```
     # Exit the program
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, 3, or 4")

print("Program Ended")
```

Unless the user chooses option 4, the menu is re-printed, and the program allows the user to select a menu option.

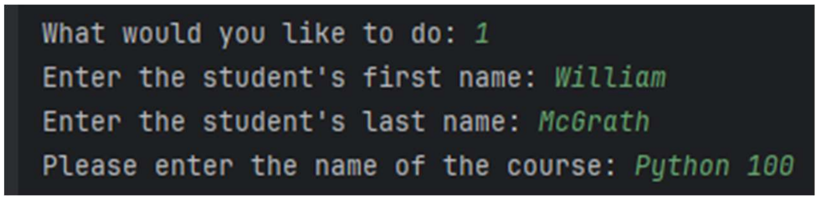
Testing

The program was tested in PyCharm and Command Prompt.

In PyCharm, two students were entered by using menu choice 1.

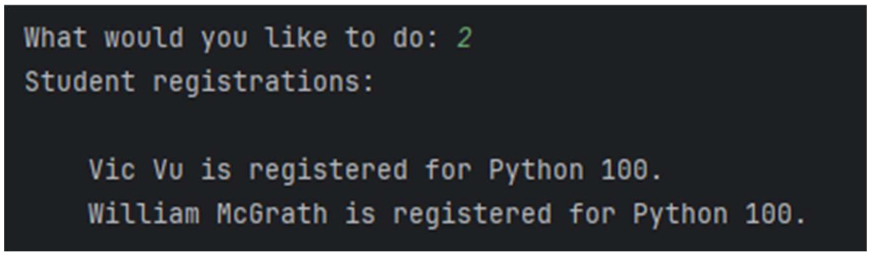


```
What would you like to do: 1
Enter the student's first name: Vic
Enter the student's last name: Vu
Please enter the name of the course: Python 100
```



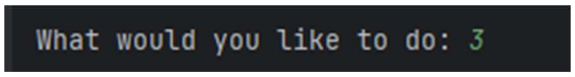
```
What would you like to do: 1
Enter the student's first name: William
Enter the student's last name: McGrath
Please enter the name of the course: Python 100
```

Next, menu choice 2 was selected to display the collected information.



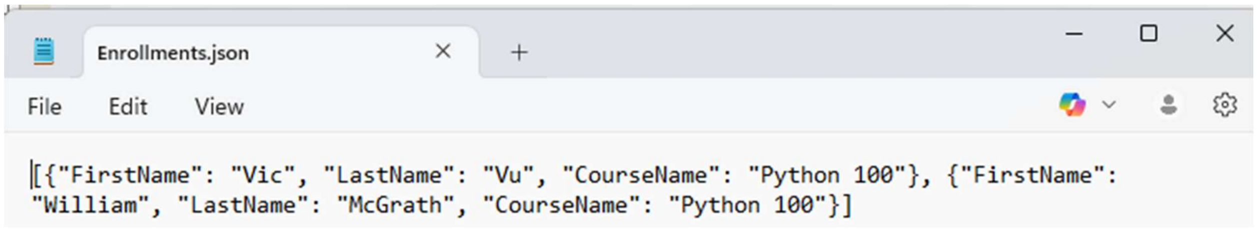
```
What would you like to do: 2
Student registrations:

    Vic Vu is registered for Python 100.
    William McGrath is registered for Python 100.
```

Next, menu option 3 was selected to display the collected information.



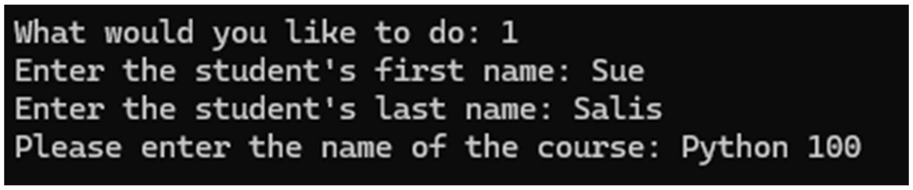
```
What would you like to do: 3
```

Enrollments.json now includes the student information.

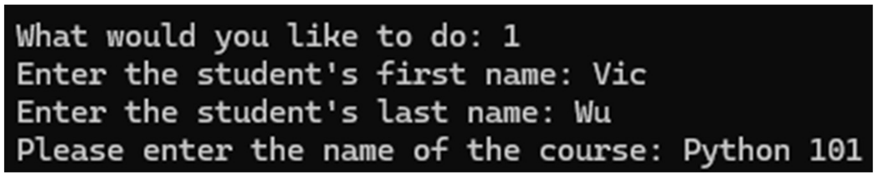


```
[{"FirstName": "Vic", "LastName": "Vu", "CourseName": "Python 100"}, {"FirstName":
"William", "LastName": "McGrath", "CourseName": "Python 100"}]
```

In Command Prompt, two students were entered by using menu choice 1.

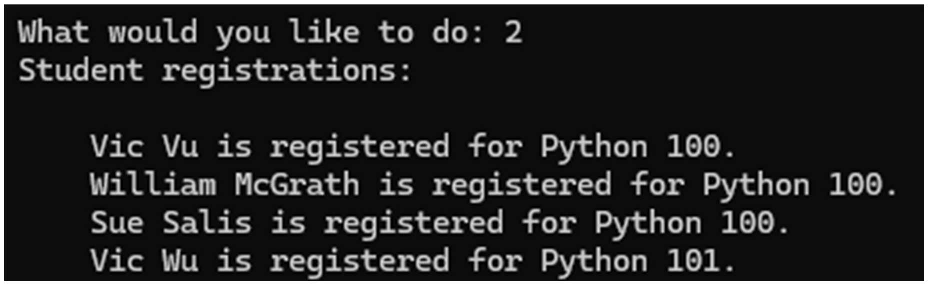


```
What would you like to do: 1
Enter the student's first name: Sue
Enter the student's last name: Salis
Please enter the name of the course: Python 100
```



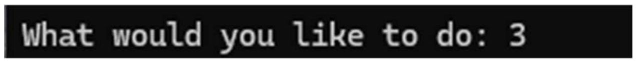
```
What would you like to do: 1
Enter the student's first name: Vic
Enter the student's last name: Wu
Please enter the name of the course: Python 101
```

Next, menu choice 2 was selected to display the collected information.



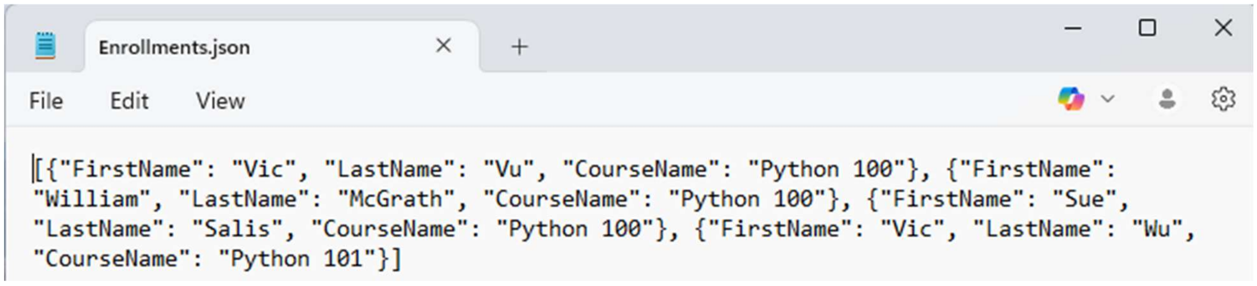
```
What would you like to do: 2
Student registrations:

    Vic Vu is registered for Python 100.
    William McGrath is registered for Python 100.
    Sue Salis is registered for Python 100.
    Vic Wu is registered for Python 101.
```

Next, menu option 3 was selected to display the collected information.



```
What would you like to do: 3
```

Enrollments.json now includes the new student information.



```
[{"FirstName": "Vic", "LastName": "Vu", "CourseName": "Python 100"}, {"FirstName":
"William", "LastName": "McGrath", "CourseName": "Python 100"}, {"FirstName": "Sue",
"LastName": "Salis", "CourseName": "Python 100"}, {"FirstName": "Vic", "LastName": "Wu",
"CourseName": "Python 101"}]
```

Summary

The Module 5 assignment was 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 Assignment04, but it adds the use of data processing using dictionaries and error handling. This document captures the steps I took to complete Assignment05.