Tellrell White

05/21/2025

Foundations of Programming: Python

Assignment 5

https://github.com/twhite320/IntroToProg-Python

Advanced Collections and Error Handling

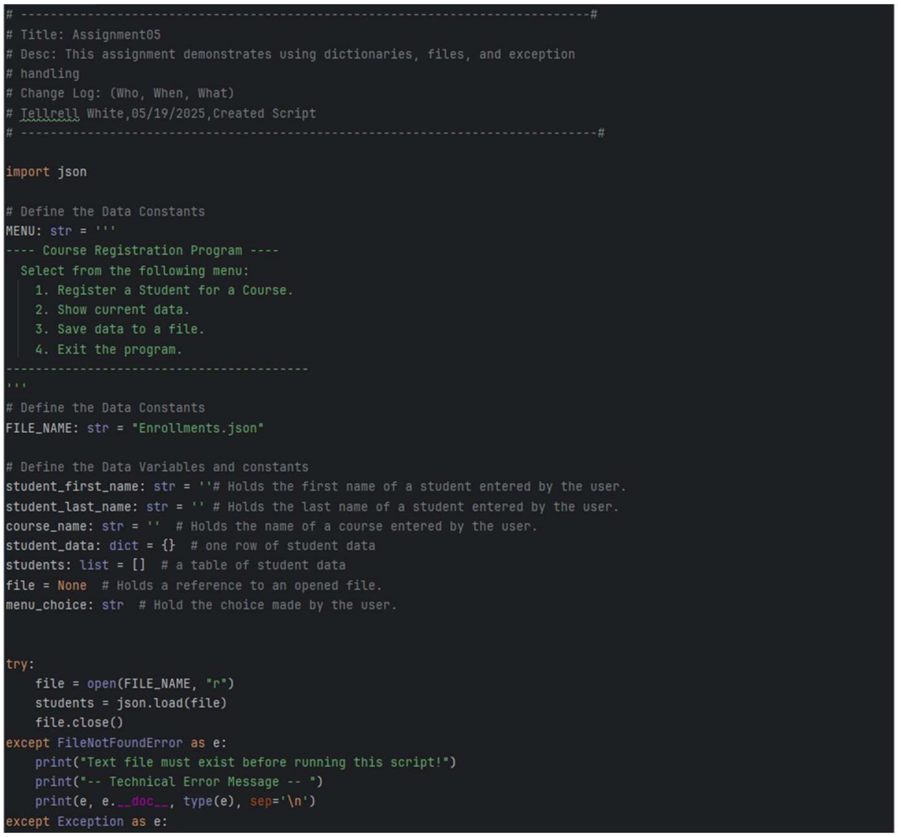
Introduction

In this document I will discuss the steps I used to create and test a Python script that displays a message about a student's registration in a Python course. This assignment is very similar to assignment04, however, the additional concepts of data processing using dictionaries and error handling were added to complete this assignment using the the Pycharm Integrated Development Environment (IDE).

Development of Python Script

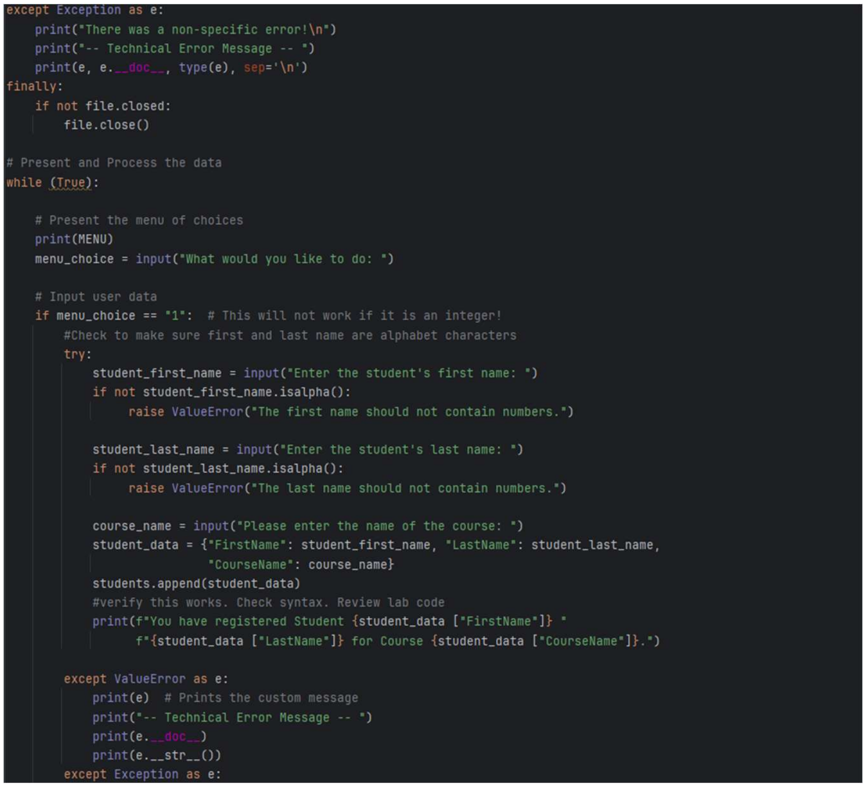
The Python script created for this assignment displayed in Figures 1, 2 and 3 was developed using the PyCharm IDE, which is an environment that allows programmers to write, debug, and manage Python projects for a host of different applications. The header section displayed at the top of the script shows my name, a short description, and a change log. After the header section, all the "setup code" including importing the "JSON" module used for file operations, and defining the constants and data variables to be used in the program are shown. Next follows the body of the script which begins with reading and storing student data into memory from the JSON file "Enrollments.json". To accomplish this, first the open() function is used to allow for reading from the JSON file. Next, the function "json.load()" reads the list of dictionaries from the JSON file and assigns them to the variable "students[]" in the program. As shown figures 1 and 2, error handling is used for reading and loading the list of dictionaries into the program. In the event of an error, such as a "FileNotFoundError" error or some other type of error, a custom message will be printed to the console along with additional information to alert the user of the error. Additionally, the error handling implemented will check to see if the file is open, and if its open it will be closed.

Next, in the script a "while loop" is presented in Figure 2 similar to the "while loop" presented in previous assignments. The "while loop" will repeat unless the user chooses to end the program by selecting "menu option 4". The "while loop" begins by printing the menu of options to the screen for the user and based on the user's selection, one of the "if-else if" conditional statements matching the user's selection will be executed. To illustrate, if the user selects menu option 1, the user will be asked to enter a first name, last name and course information which will be stored in string variables. The data in these variables will be assigned to the dictionary "student_data" and that dictionary will be appended to the list variable, "students". Error handling is also used in option 1 for inputting a first and last name. In the event that the user inputs anything other than alphabetic characters for a first or last name, or if some other error occurs while executing menu option 2, a message will be printed to the console informing the user. If the user selects option 2, a "for loop" is used to print each row of student registration data contained in the variable "lists" for the user. Selecting option 3 stores the student registration information contained in "students" to a JSON file stored in the same directory as the directory which hosts .py script, and also displays a string to the console similar to that displayed by selecting menu option 2. Error handling is also used for writing to the file as shown in Figure 3. A custom message along with additional information will be printed in the event of a formatting or some of other error. Additionally, error handling will check to if the JSON file is closed, and if not, it will be closed. Menu option 4 is selected for ending the program. Also, an "else statement" is shown in figure 2 for the scenario where the user doesn't enter a value between 1-4. In this case, the user will be prompted to re-enter a valid value.



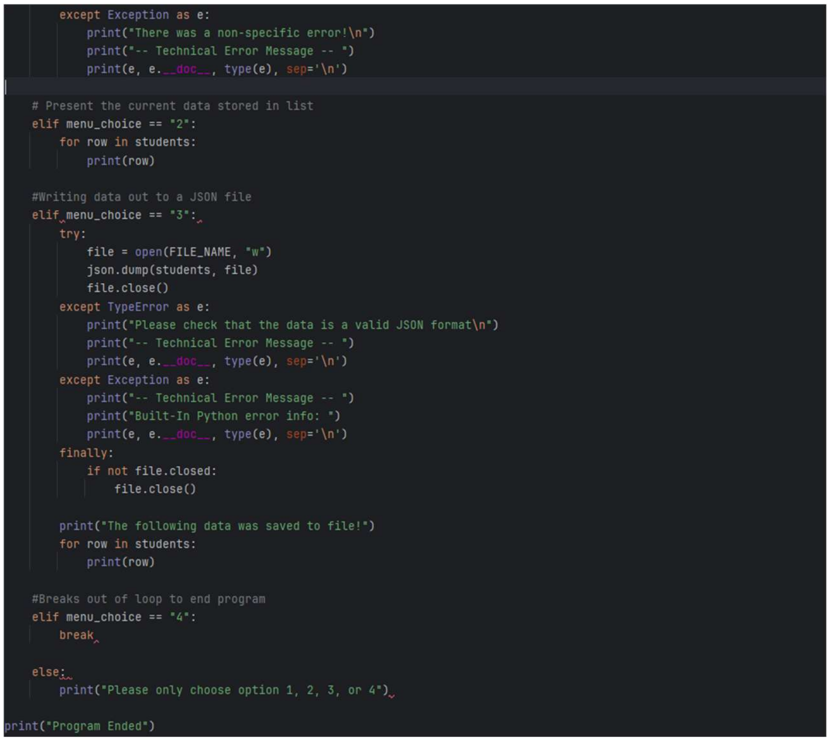
```
# ----------------------------------------------------------------------------#
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception
# handling
# Change Log: (Who, When, What)
# Tellrell White,05/19/2025,Created Script
# -----------------------------------------------------------------------------#

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 and constants
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
file = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.


try:
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError as e:
    print("Text file must exist before running this script!")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
except Exception as e:
```

Figure 1: Python Script in Pycharm



```
except Exception as e:
    print("There was a non-specific error!\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
finally:
    if not file.closed:
        file.close()

# Present and Process the data
while (True):

    # Present the menu of choices
    print(MENU)
    menu_choice = input("What would you like to do: ")

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        #Check to make sure first and last name are alphabet characters
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The first name should not contain numbers.")

            student_last_name = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The last name should not contain numbers.")

            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)
            #verify this works. Check syntax. Review lab code
            print(f"You have registered Student {student_data ["FirstName"]} "
                  f"{student_data ["LastName"]} for Course {student_data ["CourseName"]}.")

        except ValueError as e:
            print(e)  # Prints the custom message
            print("-- Technical Error Message -- ")
            print(e.__doc__)
            print(e.__str__())
        except Exception as e:
```

Figure 2: Python Script in Pycharm



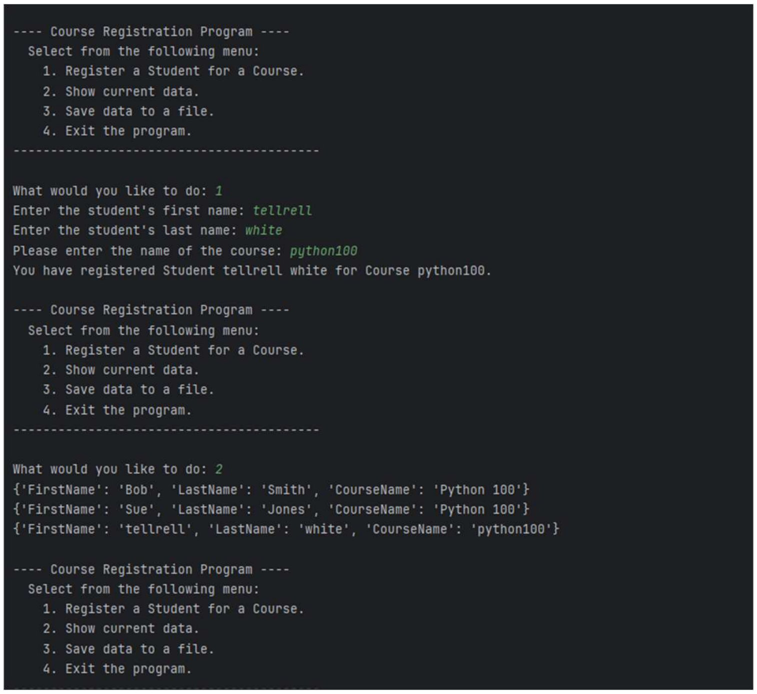
```
        except Exception as e:
            print("There was a non-specific error!\n")
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')

    # Present the current data stored in list
    elif menu_choice == "2":
        for row in students:
            print(row)

    #Writing data out to a JSON file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            json.dump(students, file)
            file.close()
        except TypeError as e:
            print("Please check that the data is a valid JSON format\n")
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')
        except Exception as e:
            print("-- Technical Error Message -- ")
            print("Built-In Python error info: ")
            print(e, e.__doc__, type(e), sep='\n')
        finally:
            if not file.closed:
                file.close()

        print("The following data was saved to file!")
        for row in students:
            print(row)

    #Breaks out of loop to end program
    elif menu_choice == "4":
        break

    else:
        print("Please only choose option 1, 2, 3, or 4")

print("Program Ended")
```

Figure 3: Python Script in Pycharm

Testing of Python Script

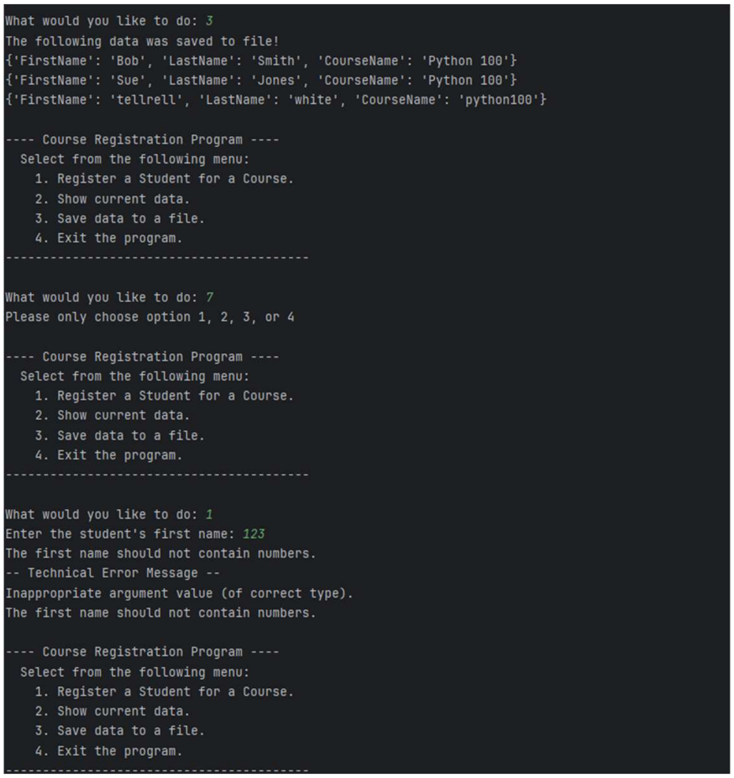
To validate that the script developed and shown in the figures above worked as intended, two methods were used to test the behavior of the script. Within the Pycharm IDE, I ran the “Assignment05.py script to not only ensure that there were no errors and that the script would run, but also to verify behavior of the script was correct. Figures 4, 5, and 6 show the output from running the “Assignment05.py” script in Pycharm. As shown in the figures, the user inputs a value matching a menu choice, and the conditional statement matching the user’s input will be executed. As shown in the figure, the user first selects “menu choice 1”, which prompts the user for a first name, last name and course information. During the second loop or iteration, the user is shown the menu again, and user selects “option 2”, which displays the registration information entered by selecting menu option 1 to the console. Next, the user selects “option 3”, which saves the user’s registration information to a JSON file and prints the student registration information stored in “students” to the screen. The contents of the original JSON file, “Enrollments.json” is shown in Figure 7 and the modified JSON file containing all the data written to the file from the “lists” variable in shown in Figure 8. During the next loop the menu is presented, and the user enters a value of “7”, which isn’t a valid choice, and the user is shown the message “Enter a valid value (1-4)” and the program restarts the loop. Next, the user selects menu choice 1 and inputs “123” when prompted for a first name. Eror handling is demonstrated by printing a message telling the user the first name cannot contain numbers along with additional information. Next, when prompted for a menu selection, the user enters the value of “4” and this ends the program.

Another method of testing the script was thru using the command terminal in windows. After launching the command terminal and navigating to the correct directory where the script exists I launched the python interpreter using the “Python” command and provided the name of the script as an argument to run it. The output of running the script is shown in Figures 9 and 10. Like the results obtained from Pycharm, when selecting menu choices 1 thru 4, the program displayed correct results.



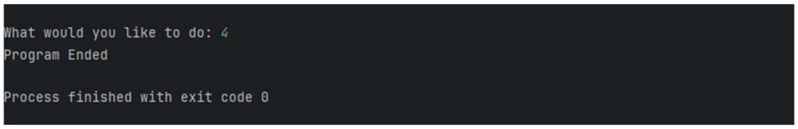
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------

What would you like to do: 1
Enter the student's first name: tellrell
Enter the student's last name: white
Please enter the name of the course: python100
You have registered Student tellrell white for Course python100.

---- 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
{'FirstName': 'Bob', 'LastName': 'Smith', 'CourseName': 'Python 100'}
{'FirstName': 'Sue', 'LastName': 'Jones', 'CourseName': 'Python 100'}
{'FirstName': 'tellrell', 'LastName': 'white', 'CourseName': 'python100'}

---- 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: Output from Pycharm



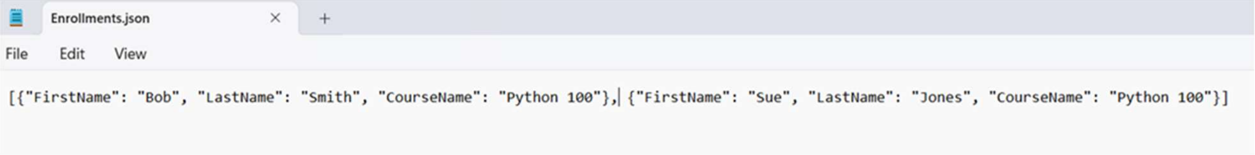
```
What would you like to do: 3
The following data was saved to file!
{'FirstName': 'Bob', 'LastName': 'Smith', 'CourseName': 'Python 100'}
{'FirstName': 'Sue', 'LastName': 'Jones', 'CourseName': 'Python 100'}
{'FirstName': 'tellrell', 'LastName': 'white', 'CourseName': 'python100'}

---- 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: 7
Please only choose option 1, 2, 3, or 4

---- 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: 123
The first name should not contain numbers.
-- Technical Error Message --
Inappropriate argument value (of correct type).
The first name should not contain numbers.

---- 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 5: Output in Pycharm



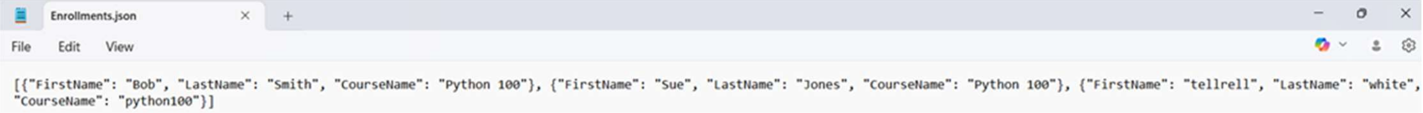
```
What would you like to do: 4
Program Ended

Process finished with exit code 0
```

Figure 6: Output from Pycharm



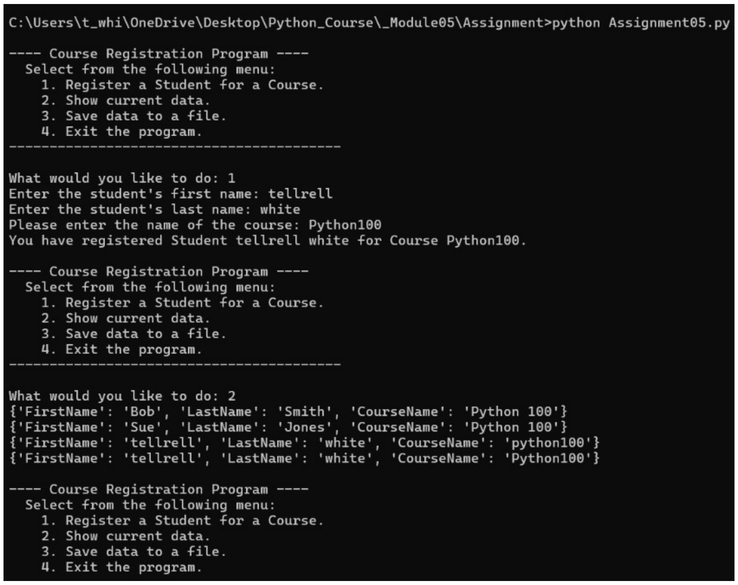
```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"}, {"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"}]
```

Figure 7: Original JSON File



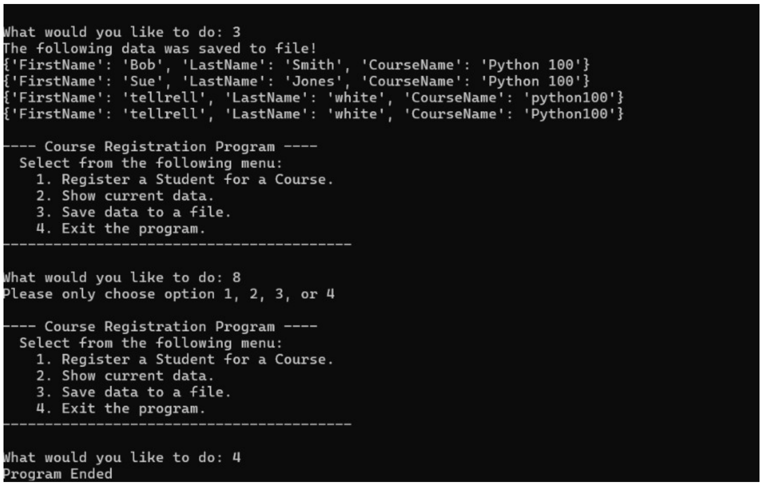
```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"}, {"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"}, {"FirstName": "tellrell", "LastName": "white",
"CourseName": "python100"}]
```

Figure 8: Modified JSON File



```
C:\Users\t_whi\OneDrive\Desktop\Python_Course\_Module05\Assignment>python Assignment05.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: tellrell
Enter the student's last name: white
Please enter the name of the course: Python100
You have registered Student tellrell white for Course Python100.

---- 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
{'FirstName': 'Bob', 'LastName': 'Smith', 'CourseName': 'Python 100'}
{'FirstName': 'Sue', 'LastName': 'Jones', 'CourseName': 'Python 100'}
{'FirstName': 'tellrell', 'LastName': 'white', 'CourseName': 'python100'}
{'FirstName': 'tellrell', 'LastName': 'white', 'CourseName': 'Python100'}

---- 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 9: Windows Terminal Output



```
What would you like to do: 3
The following data was saved to file!
{'FirstName': 'Bob', 'LastName': 'Smith', 'CourseName': 'Python 100'}
{'FirstName': 'Sue', 'LastName': 'Jones', 'CourseName': 'Python 100'}
{'FirstName': 'tellrell', 'LastName': 'white', 'CourseName': 'python100'}
{'FirstName': 'tellrell', 'LastName': 'white', 'CourseName': 'Python100'}

---- 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: 8
Please only choose option 1, 2, 3, or 4

---- 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 10: Windows Terminal Output

Summary

As discussed in this document, a Python script was developed utilizing the concepts of data processing using dictionaries and error handling files learned in module 5 to display to a user information about registration into a Python course. This script was tested using Pycharm, the terminal in windows, and examination of a JSON file for correct behavior.