Olivier Richer
May 26, 2025, IT FDN 110 A Sp 25: Foundations of Programming: Python
GitHub hyperlink: olik2000 · GitHub
Assignment 05- Advanced Collections and Error Handling.

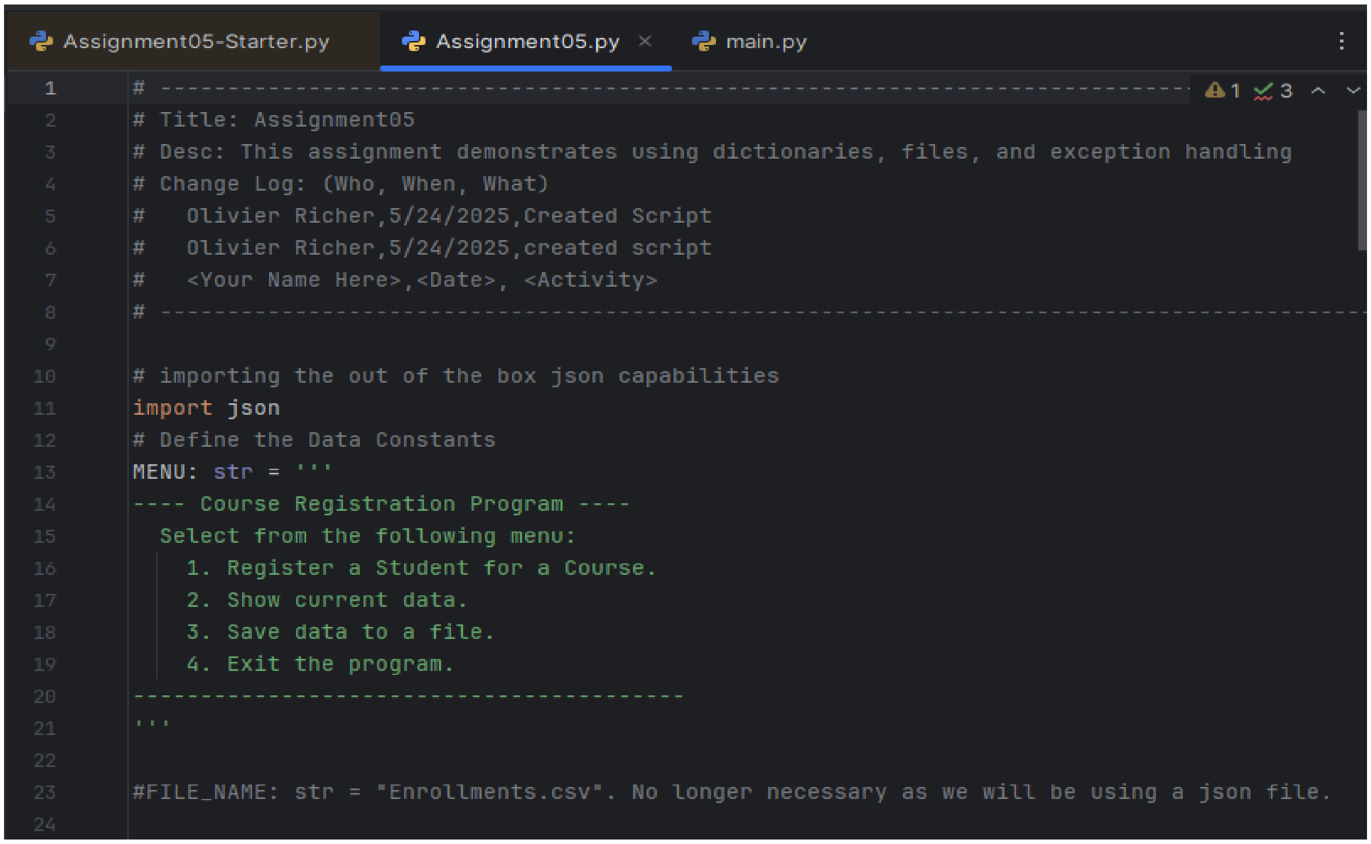
Dictionary and exception handling.

Introduction:

Assignment #5 scope of work is very similar to assignment#04. However, it adds It adds the use of data processing using dictionaries and exception handling**.** Lastly, this report will be divided into three parts (the script, the execution of the program, and the debugging).

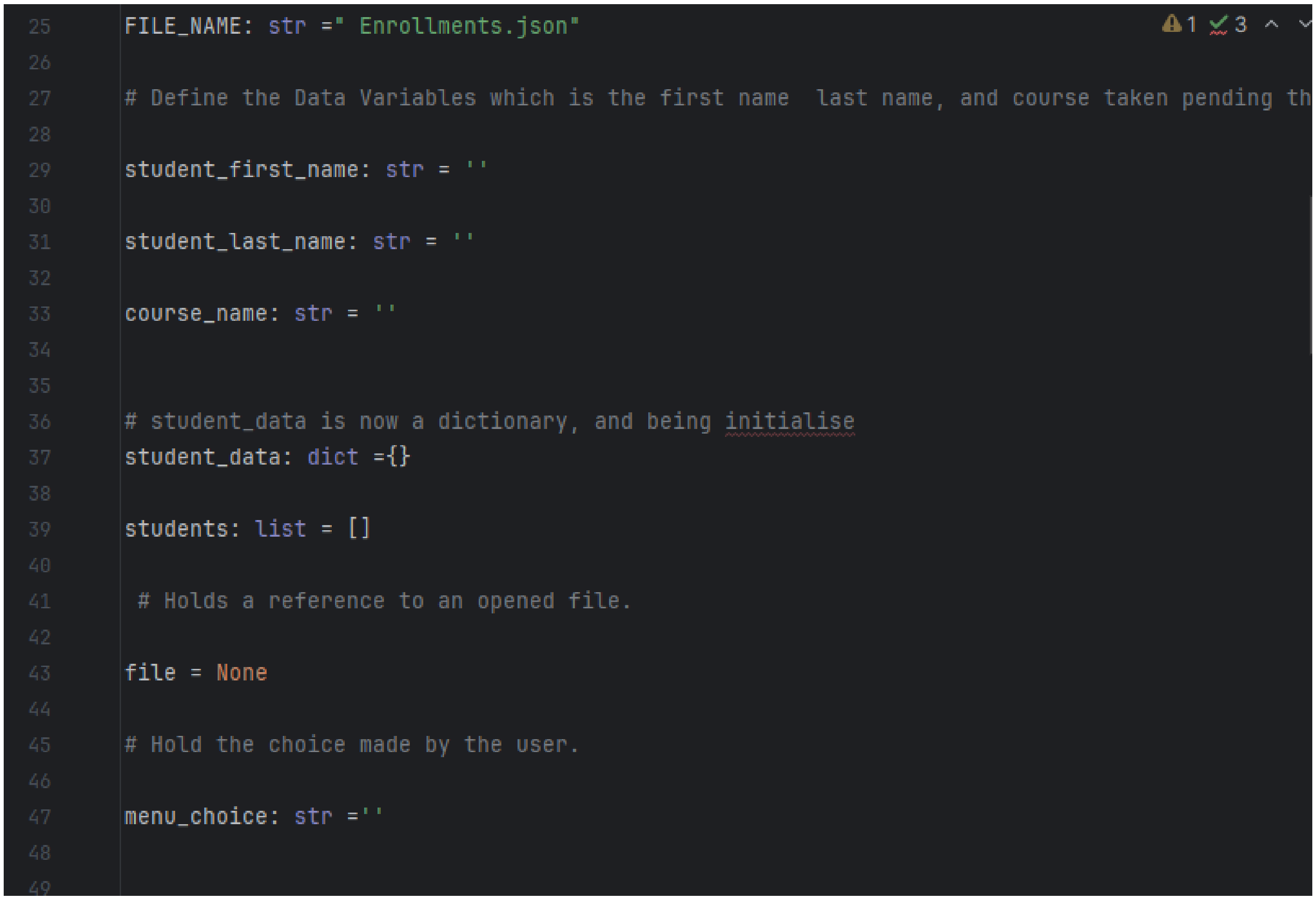
1. The script.

The departure from assignment#04 is that we are using the Json library, and Json files. As such we are using commands associated with handling Json files. Also, we are using dictionary which is different type of structure/object in which each "Key" is associated with a "Value". In the previous assignment we were using a list which was using indexation. Lastly, exception handling command such as "try", "except", and "finally", encapsulate block of codes/script for which we can custom some errors and handle it rather than either let he program stop or let the user doing some "forbidding" operation (see fig1 through fig6 for the script).



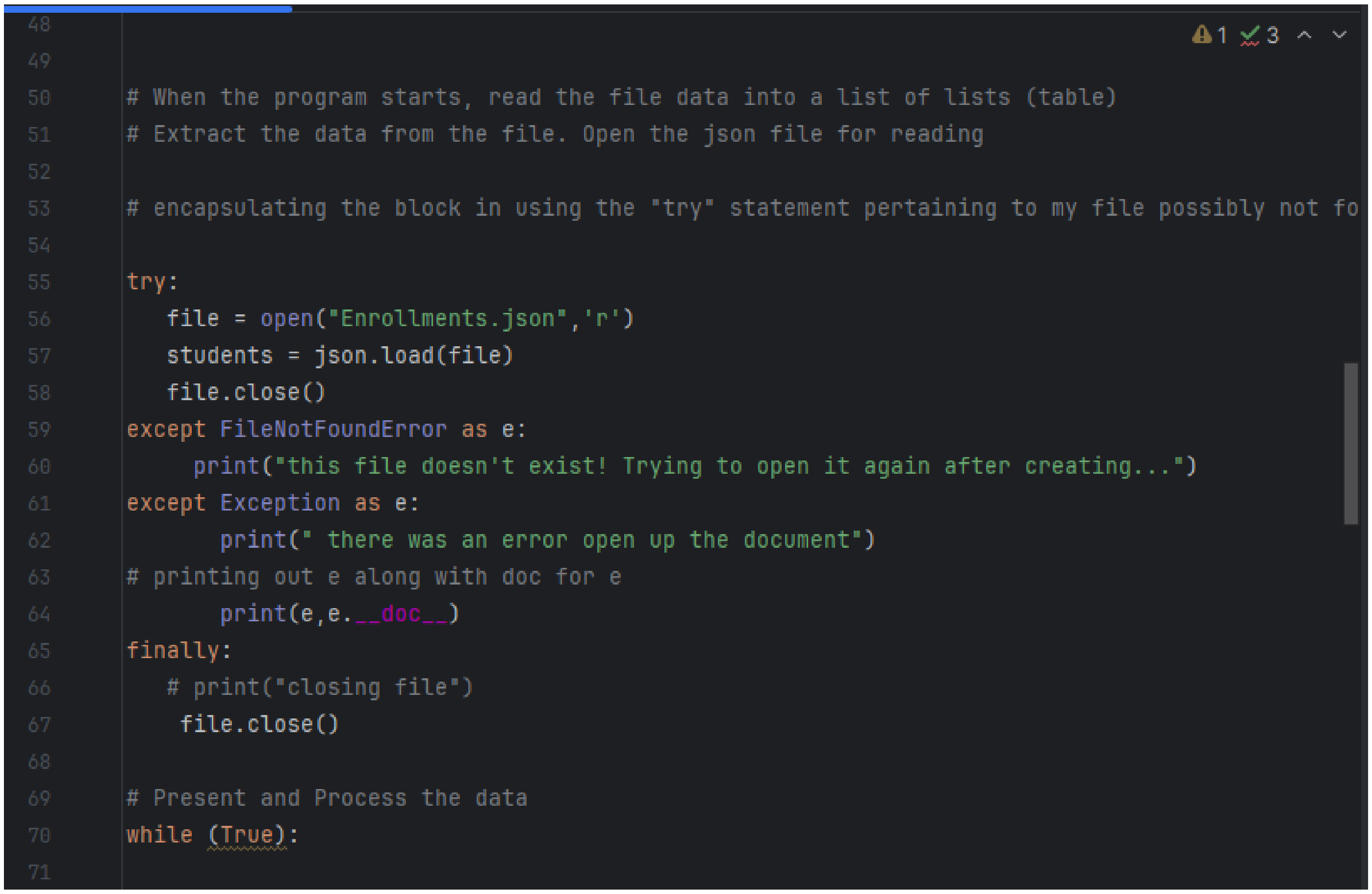
```
# -----------------------------------------------------------------------------
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   Olivier Richer,5/24/2025,Created Script
#   Olivier Richer,5/24/2025,created script
#   <Your Name Here>,<Date>, <Activity>
# -----------------------------------------------------------------------------

# importing the out of the box json capabilities
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.
-----------------------------------------
'''

#FILE_NAME: str = "Enrollments.csv". No longer necessary as we will be using a json file.
```

Fig1



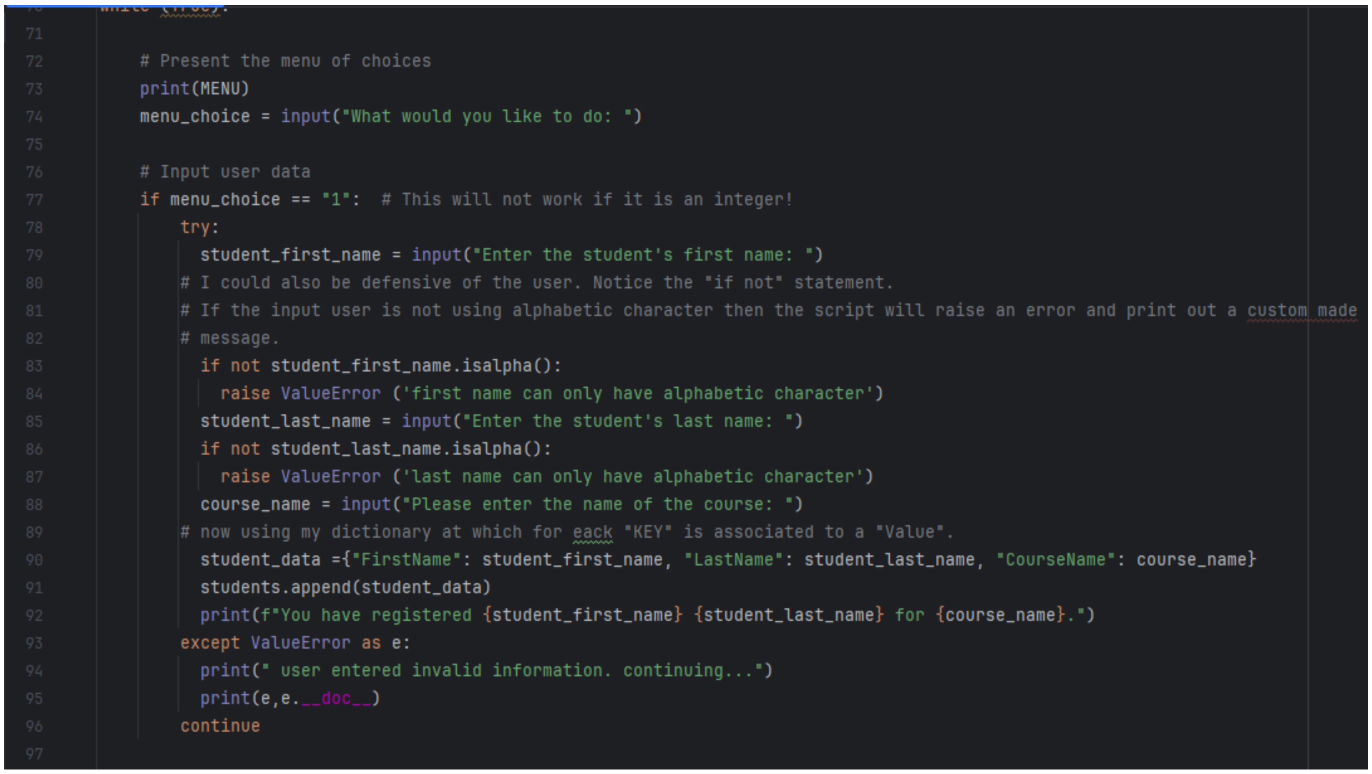
```
FILE_NAME: str =" Enrollments.json"

# Define the Data Variables which is the first name  last name, and course taken pending th

student_first_name: str = ''

student_last_name: str = ''

course_name: str = ''


# student_data is now a dictionary, and being initialise
student_data: dict ={}

students: list = []

 # Holds a reference to an opened file.

file = None

# Hold the choice made by the user.

menu_choice: str =''

```

Fig2



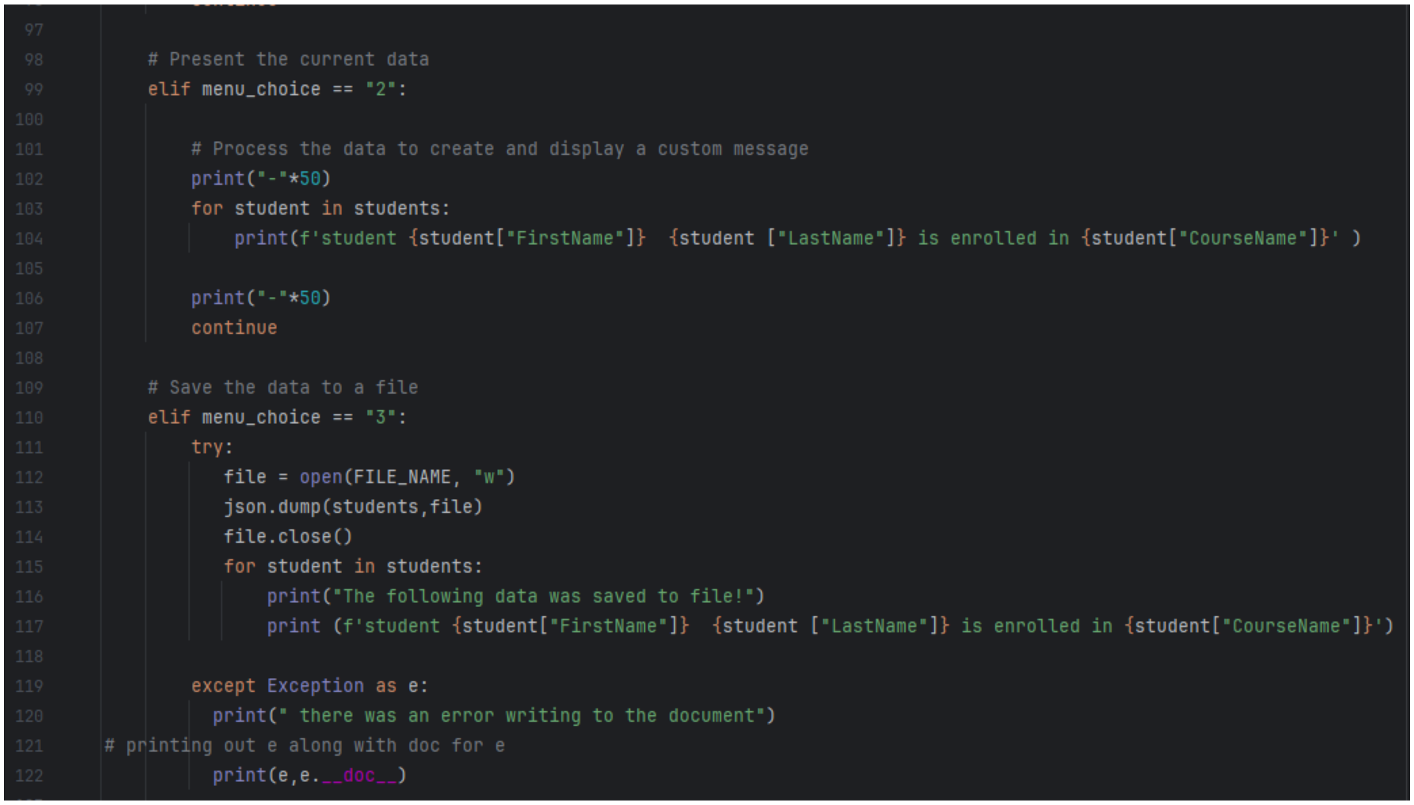
```


# When the program starts, read the file data into a list of lists (table)
# Extract the data from the file. Open the json file for reading

# encapsulating the block in using the "try" statement pertaining to my file possibly not fo

try:
   file = open("Enrollments.json",'r')
   students = json.load(file)
   file.close()
except FileNotFoundError as e:
     print("this file doesn't exist! Trying to open it again after creating...")
except Exception as e:
       print(" there was an error open up the document")
# printing out e along with doc for e
       print(e,e.__doc__)
finally:
   # print("closing file")
    file.close()

# Present and Process the data
while (True):

```

Fig3



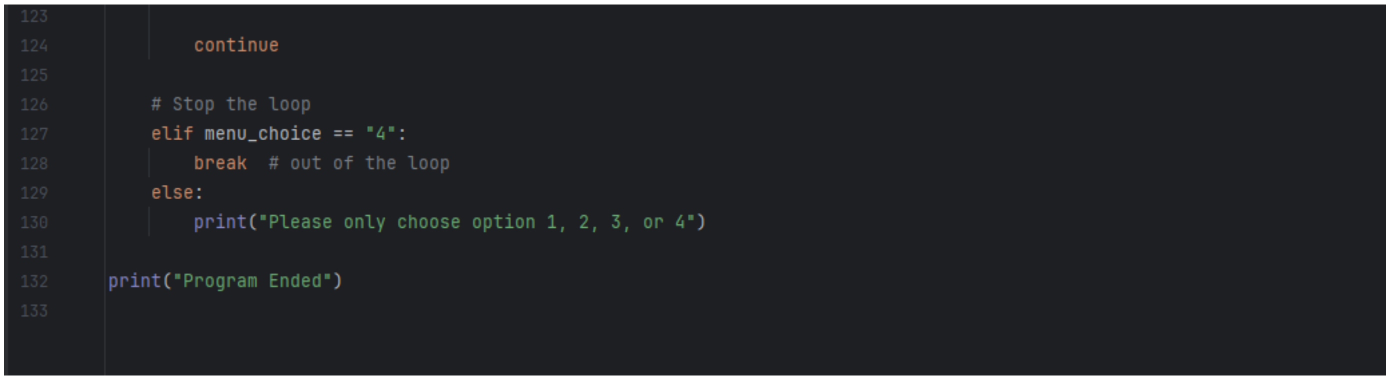
```python

    # 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!
        try:
            student_first_name = input("Enter the student's first name: ")
        # I could also be defensive of the user. Notice the "if not" statement.
        # If the input user is not using alphabetic character then the script will raise an error and print out a custom made
        # message.
            if not student_first_name.isalpha():
                raise ValueError ('first name can only have alphabetic character')
            student_last_name = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError ('last name can only have alphabetic character')
            course_name = input("Please enter the name of the course: ")
        # now using my dictionary at which for eack "KEY" is associated to a "Value".
            student_data ={"FirstName": student_first_name, "LastName": student_last_name, "CourseName": course_name}
            students.append(student_data)
            print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
        except ValueError as e:
            print(" user entered invalid information. continuing...")
            print(e,e.__doc__)
        continue

```

Fig4



```python

    # Present the current data
    elif menu_choice == "2":

        # Process the data to create and display a custom message
        print("-"*50)
        for student in students:
            print(f'student {student["FirstName"]}  {student ["LastName"]} is enrolled in {student["CourseName"]}' )

        print("-"*50)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        try:
           file = open(FILE_NAME, "w")
           json.dump(students,file)
           file.close()
           for student in students:
               print("The following data was saved to file!")
               print (f'student {student["FirstName"]}  {student ["LastName"]} is enrolled in {student["CourseName"]}')

        except Exception as e:
          print(" there was an error writing to the document")
# printing out e along with doc for e
          print(e,e.__doc__)
```

Fig5



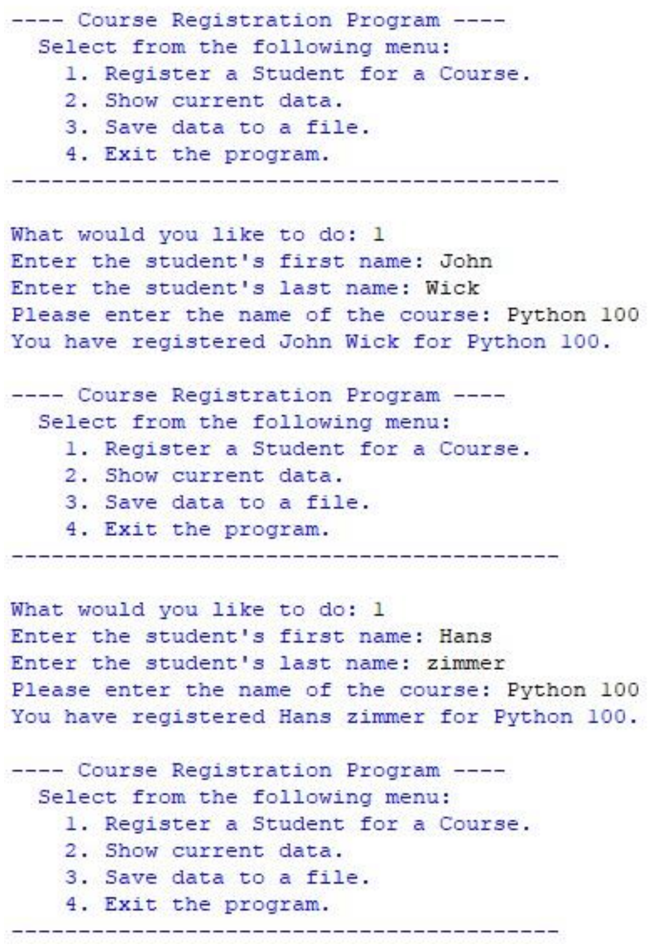
```python

        continue

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

print("Program Ended")

```

Fig6.

2. Running/Executing the script.

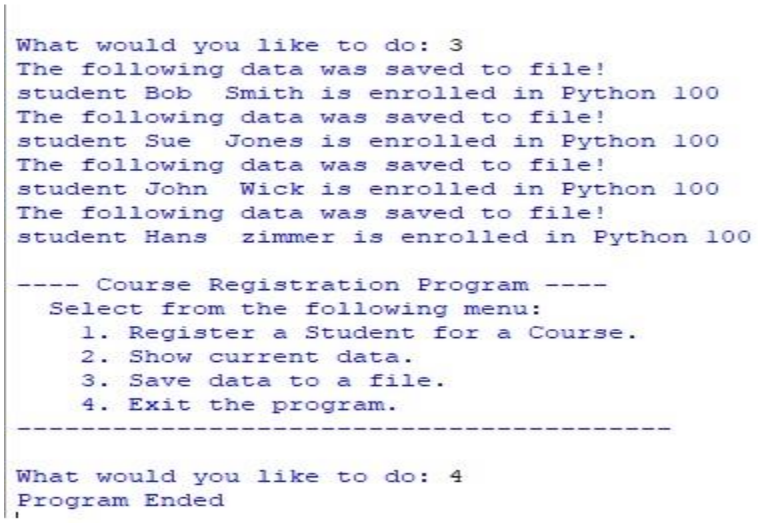
Here we will be running multiple cases to test the script and the custom errors that we design.
First, we will show the “well behave” script/user. Second, the json file is empty/not existing. Third, the user does not use alphabetic characters for name and last name. And finally, pending choice 3, having issue with the file not loading (see fig 7 through fig 15).

a) Well behave case.



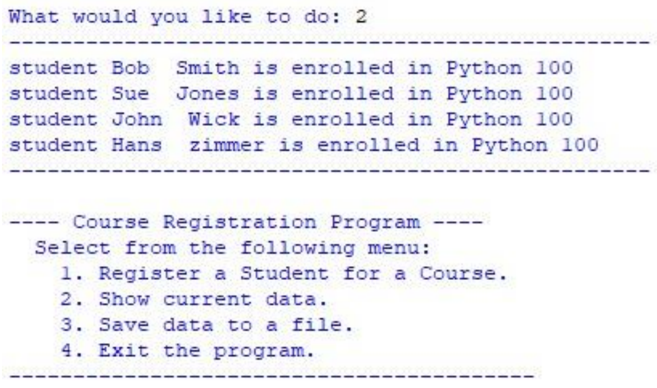
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------

What would you like to do: 1
Enter the student's first name: John
Enter the student's last name: Wick
Please enter the name of the course: Python 100
You have registered John Wick 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: 1
Enter the student's first name: Hans
Enter the student's last name: zimmer
Please enter the name of the course: Python 100
You have registered Hans zimmer 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.
-----------------------------------------
```

Fig7.



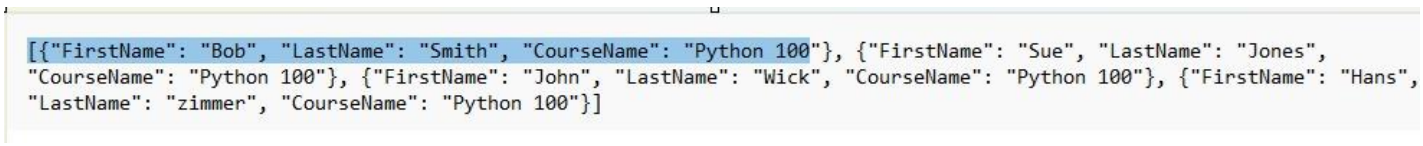
```
What would you like to do: 3
The following data was saved to file!
student Bob  Smith is enrolled in Python 100
The following data was saved to file!
student Sue  Jones is enrolled in Python 100
The following data was saved to file!
student John  Wick is enrolled in Python 100
The following data was saved to file!
student Hans  zimmer 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: 4
Program Ended
```

Fig8



```
What would you like to do: 2
--------------------------------------------------
student Bob  Smith is enrolled in Python 100
student Sue  Jones is enrolled in Python 100
student John  Wick is enrolled in Python 100
student Hans  zimmer 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.
-----------------------------------------
```

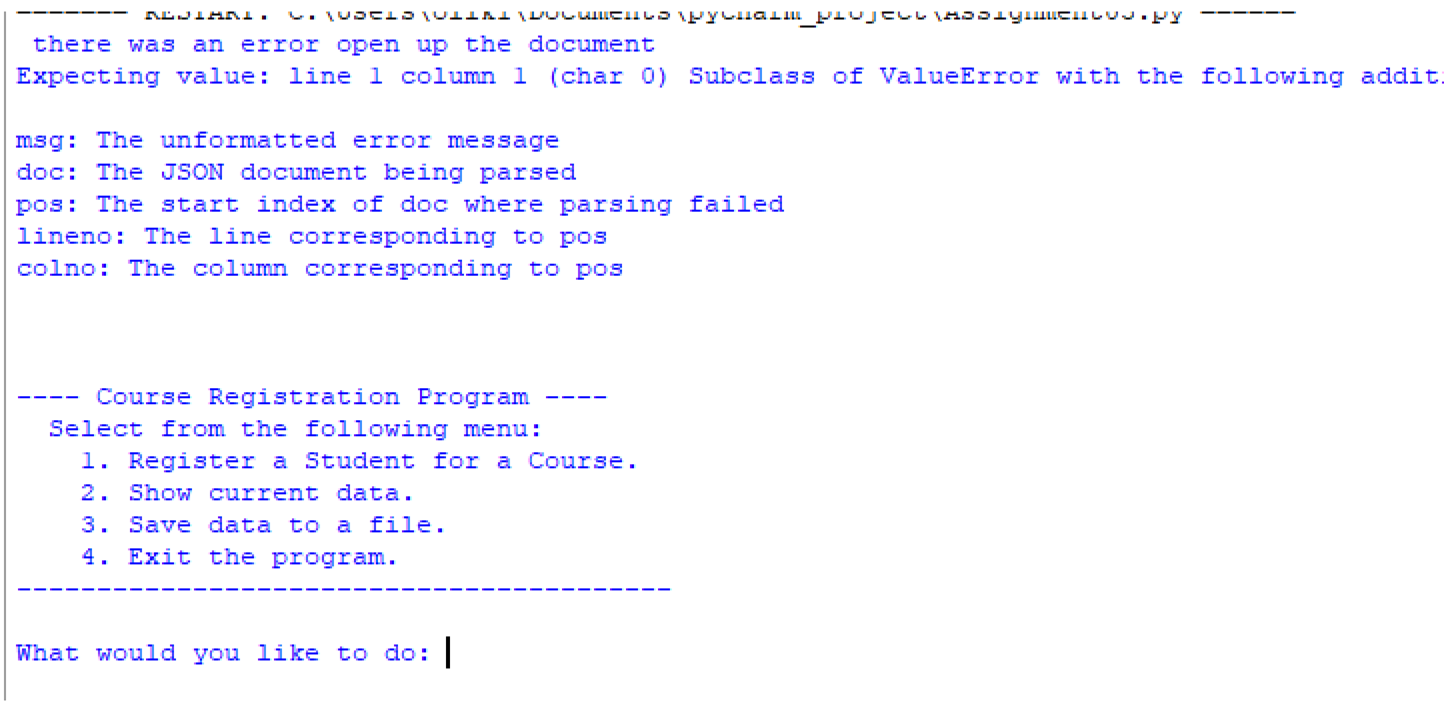
Fig9



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

Fig10

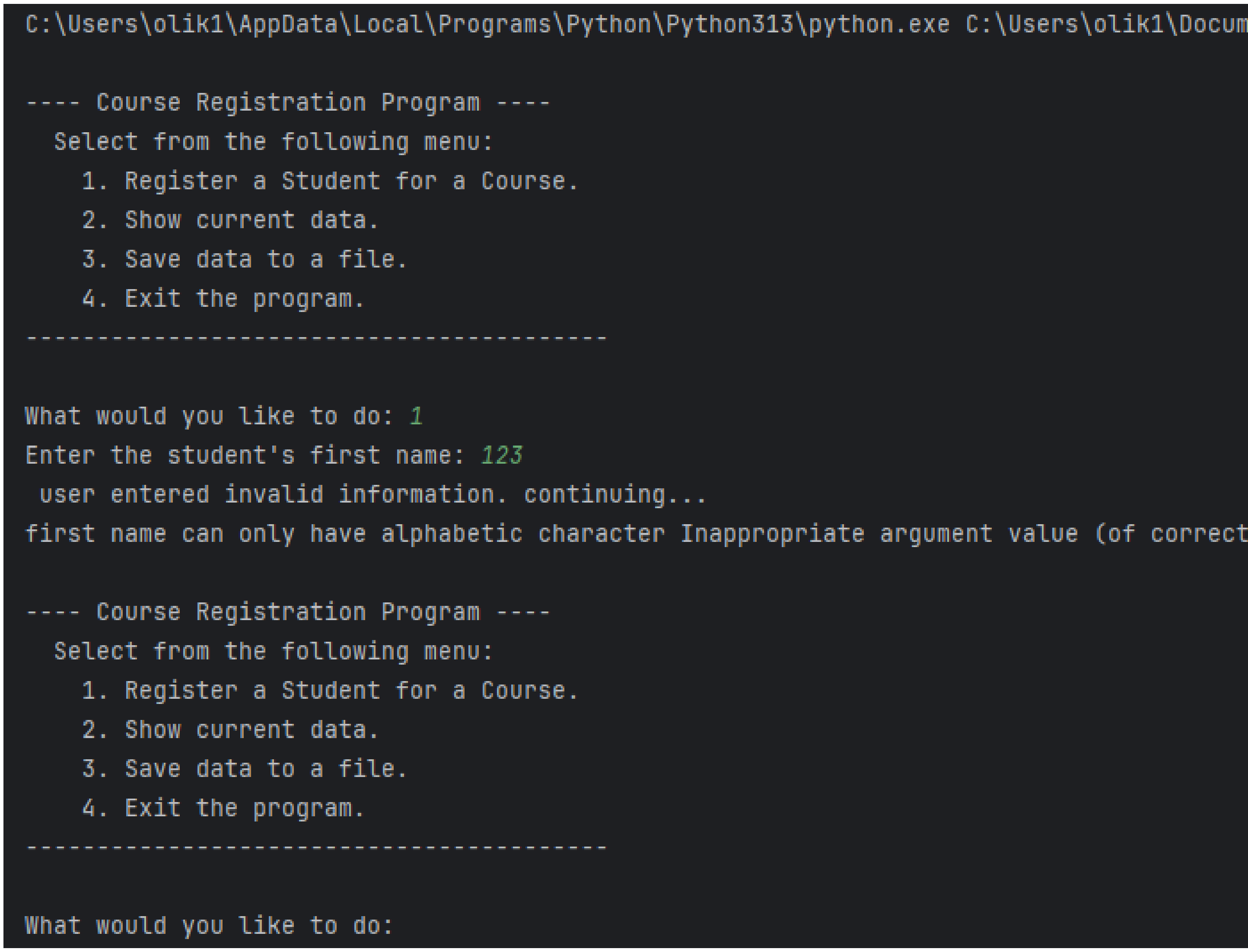
b) <u>Case json file is empty/not existing.</u>



```
------- RESTART: C:\Users\olik1\Documents\pycharm_project\Assignment05.py ------
 there was an error open up the document
Expecting value: line 1 column 1 (char 0) Subclass of ValueError with the following addit

msg: The unformatted error message
doc: The JSON document being parsed
pos: The start index of doc where parsing failed
lineno: The line corresponding to pos
colno: The column corresponding to pos



---- 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: 
```

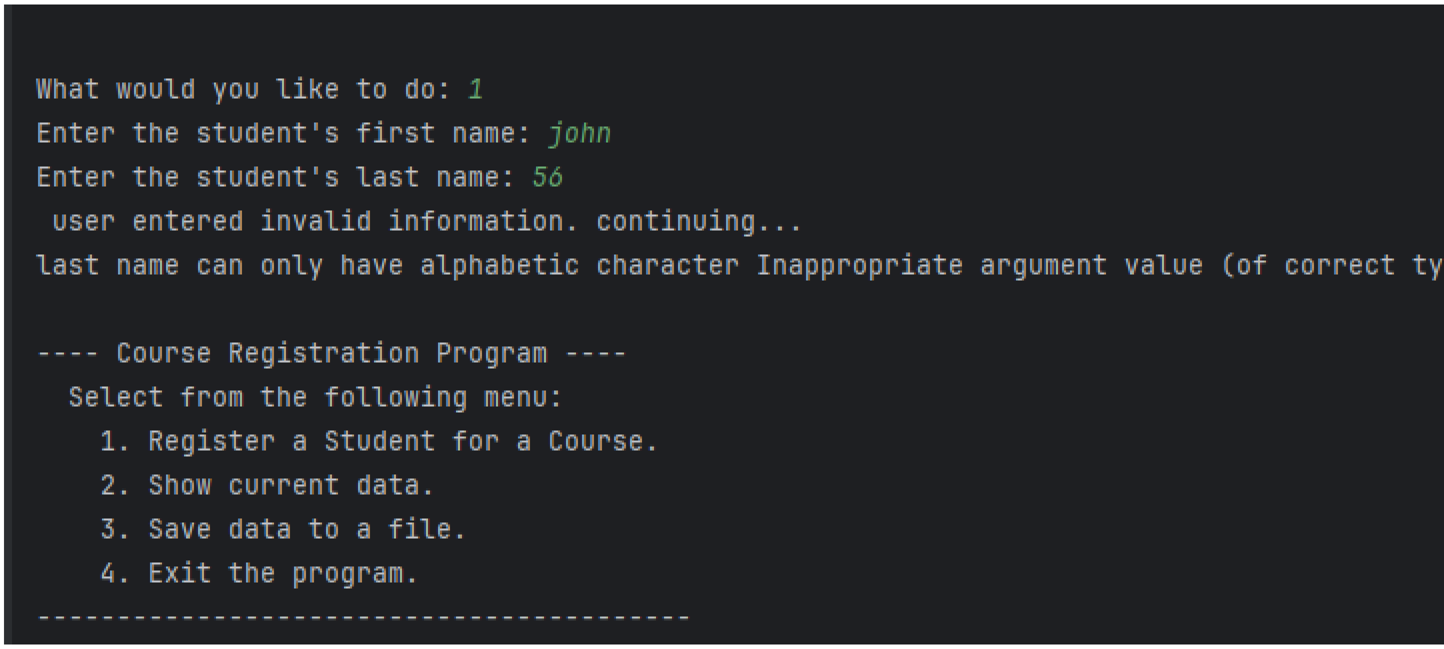
Fig 11.

c) Case: the user does not use alphabetic characters for name and last name.



```
C:\Users\olik1\AppData\Local\Programs\Python\Python313\python.exe C:\Users\olik1\Docum

---- 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
 user entered invalid information. continuing...
first name can only have alphabetic character Inappropriate argument value (of correct

---- 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:
```

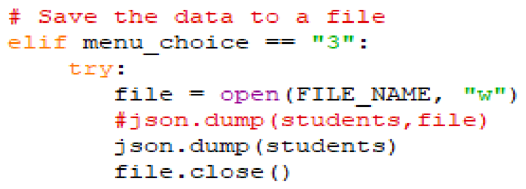
Fig12



```
What would you like to do: 1
Enter the student's first name: john
Enter the student's last name: 56
 user entered invalid information. continuing...
last name can only have alphabetic character Inappropriate argument value (of correct ty

---- 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.
-----------------------------------------
```

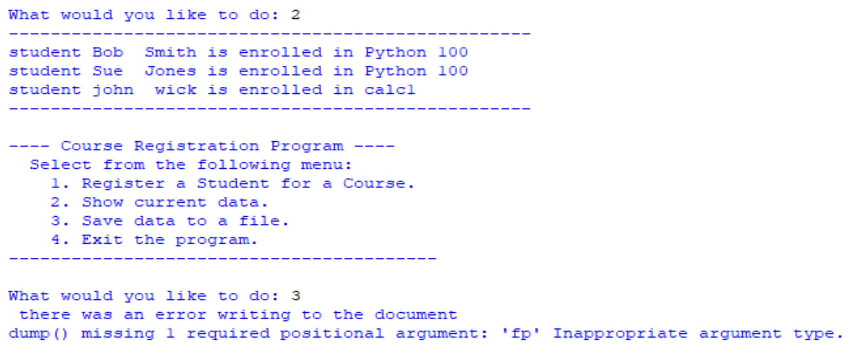
Fig13.

d) case pending choice 3, having issue with the file not loading.



```
# Save the data to a file
elif menu_choice == "3":
    try:
       file = open(FILE_NAME, "w")
       #json.dump(students,file)
       json.dump(students)
       file.close()
```

Fig14



```
What would you like to do: 2
--------------------------------------------------
student Bob  Smith is enrolled in Python 100
student Sue  Jones is enrolled in Python 100
student john  wick is enrolled in calc1
--------------------------------------------------

---- 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
 there was an error writing to the document
dump() missing 1 required positional argument: 'fp' Inappropriate argument type.
```

Fig15.

3. The debugging.

Below are the following lessons I learnt:

- Jason does not mean json. I am sure that everyone will understand the humor.
- Check your json file. I did not, and used different key names and spent a lot of times wondering what I did wrong before figuring out that firstname is different than FirstName. I believe that there is a lesson learnt right there.
- So many issues with indentation. I should know better by now.
- In doing iterations, it is easy to get confused with dealing with more structure
- Exception handling does require some maturity with programming in python which will be acquired over time. I am not yet there.

Conclusion.

In this assignment, one is dealing with json files which appear to have far more capabilities than CVS files.
Also, we used the dictionary command that offers a way to deal with manipulating data in using a key-value mechanism rather than indexation as previously seen with list. Lastly, we made custom error condition and allowed us to be defensive about the users choices that we can handle using the commands try, except, and finally. The error-handling part is probably the one which requires some experience in knowing which one is relevant.