Ismail Baha Eldin

November 013, 2024

Introduction to Programming with Python

Assignment 05

https://github.com/7for7/IntroProgPython-Mod05

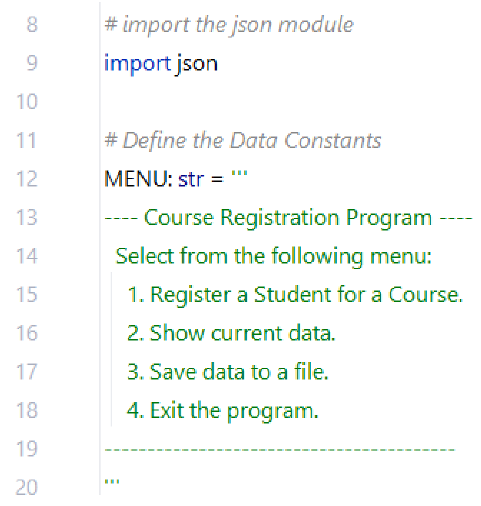
Dictionaries and Error Handling

Introduction

This assignment teaches how to process data using JSON files and error handling. The script demonstrates the use of constants, variables, and dictionaries in capturing data through a user menu, displays the results in a list of dictionaries format, and stores the results in a JSON data file.

The Program Code

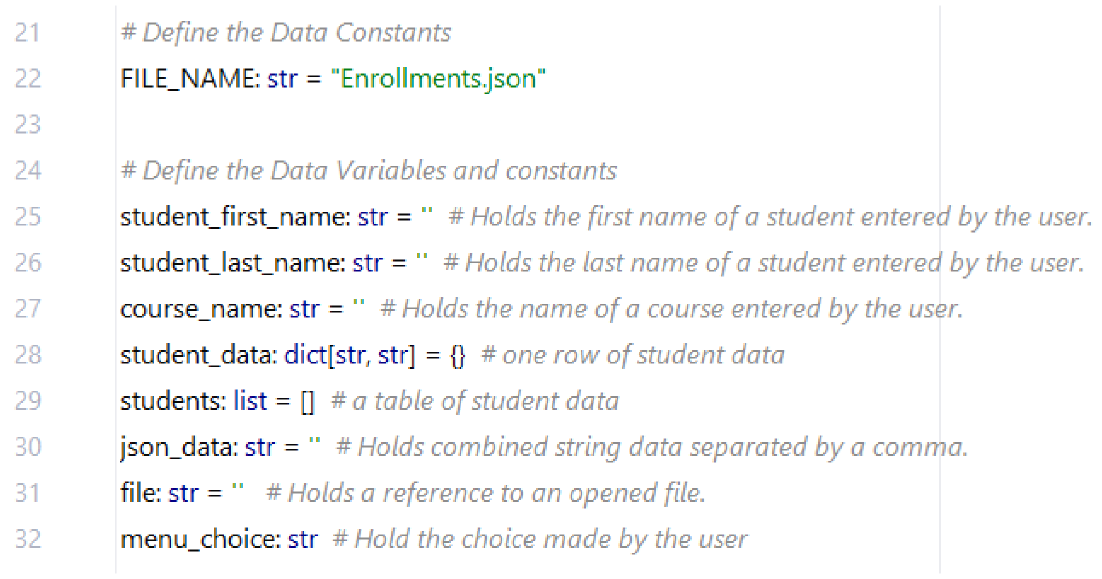
The program code starts with the script header. It uses comments throughout to declare and define steps. The JSON module is imported into Pycharm and the selection menu shows the available constant user options where selections are stored as strings (Figure 1).



```python
# import the json module
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.
-----------------------------------------
'''
```

Figure 1: User menu options

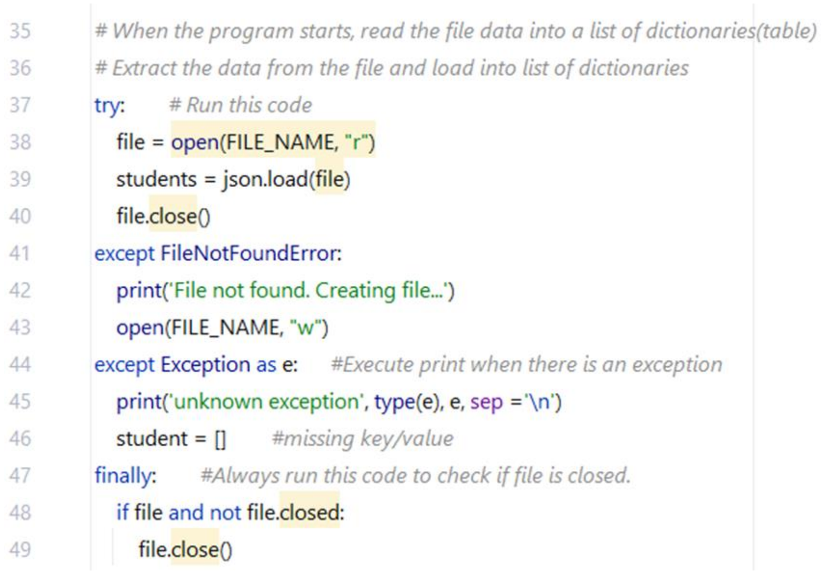
The file, input, list, and dictionary variables are defined as sting variables and initialized. The JSON file exists and is declared as ‘Enrollments.json’ (Figure 2).



```
# 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[str, str] = {}  # one row of student data
students: list = []  # a table of student data
json_data: str = ''  # Holds combined string data separated by a comma.
file: str = ''   # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user
```

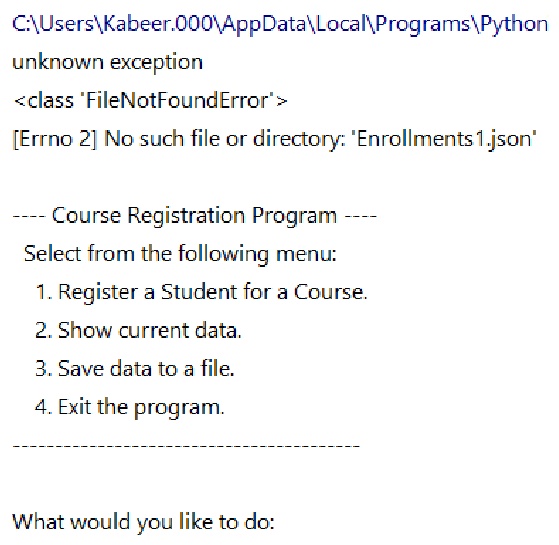
Figure 2: User menu options

The script opens and reads the contents of the file Enrollments.json containing students’ information as key/value pairs enclosed in braces separated by commas. The JSON load() command reads the file data in the ‘students’ list of dictionary items. The try-except block handles errors associated with a missing file, declares the opening of a file with the same name, and finally ensures file closure (Figure 3).



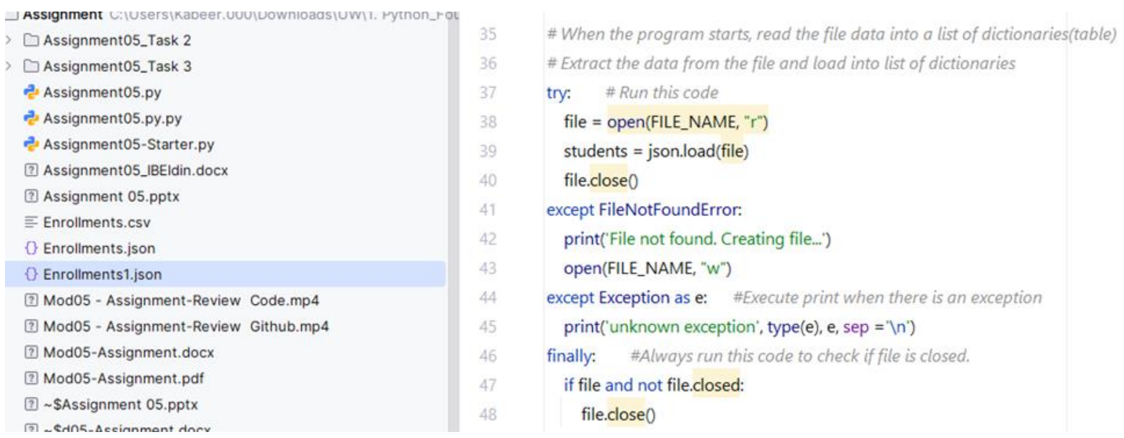
```
# When the program starts, read the file data into a list of dictionaries(table)
# Extract the data from the file and load into list of dictionaries
try:      # Run this code
   file = open(FILE_NAME, "r")
   students = json.load(file)
   file.close()
except FileNotFoundError:
   print('File not found. Creating file...')
   open(FILE_NAME, "w")
except Exception as e:      #Execute print when there is an exception
   print('unknown exception', type(e), e, sep ='\n')
   student = []      #missing key/value
finally:      #Always run this code to check if file is closed.
   if file and not file.closed:
      file.close()
```

Figure 3: Open and read JSON file student data



```
C:\Users\Kabeer.000\AppData\Local\Programs\Python
unknown exception
<class 'FileNotFoundError'>
[Errno 2] No such file or directory: 'Enrollments1.json'

---- 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:
```

Figure 4: Error handling if Enrollments1.json file not exist

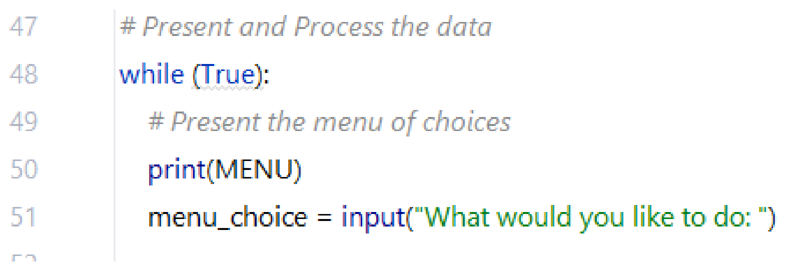


```
Assignment C:\Users\Kabeer.000\Downloads\UW\1. Python_Fo
  Assignment05_Task 2
  Assignment05_Task 3
  Assignment05.py
  Assignment05.py.py
  Assignment05-Starter.py
  Assignment05_IBEldin.docx
  Assignment 05.pptx
  Enrollments.csv
  Enrollments.json
  Enrollments1.json
  Mod05 - Assignment-Review  Code.mp4
  Mod05 - Assignment-Review  Github.mp4
  Mod05-Assignment.docx
  Mod05-Assignment.pdf
  ~$Assignment 05.pptx
```

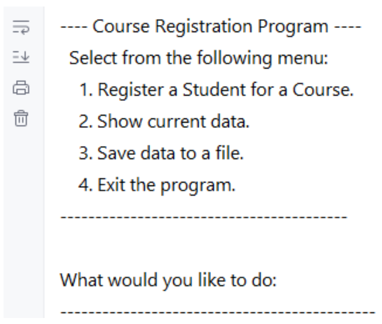
```python
# When the program starts, read the file data into a list of dictionaries(table)
# Extract the data from the file and load into list of dictionaries
try:        # Run this code
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError:
    print('File not found. Creating file...')
    open(FILE_NAME, "w")
except Exception as e:     #Execute print when there is an exception
    print('unknown exception', type(e), e, sep ='\n')
finally:      #Always run this code to check if file is closed.
    if file and not file.closed:
        file.close()
```

Figure 5: Creation of Enrollments1.json if file not exist

A While() loop is used to display the menu for an input of four choices (Figure 6 & 7).



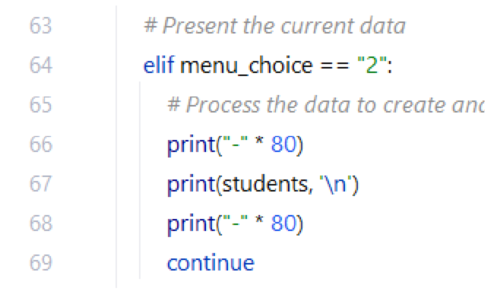
```python
# Present and Process the data
while (True):
    # Present the menu of choices
    print(MENU)
    menu_choice = input("What would you like to do: ")
```



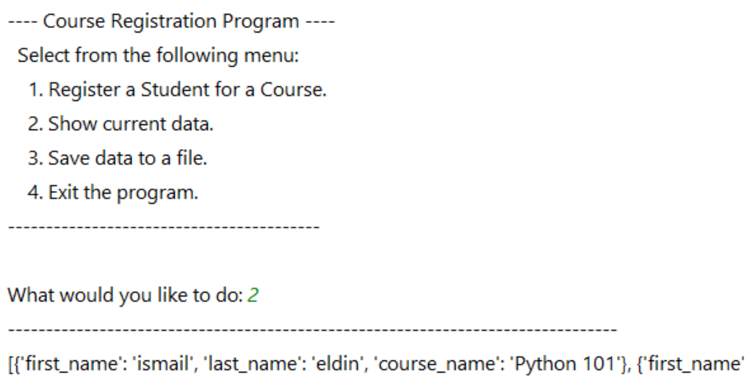
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------

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

Figure 6 & 7 : Menu selection

On menu choice 2, the data read from the JSON file is displyed (Figure 8 and 9)



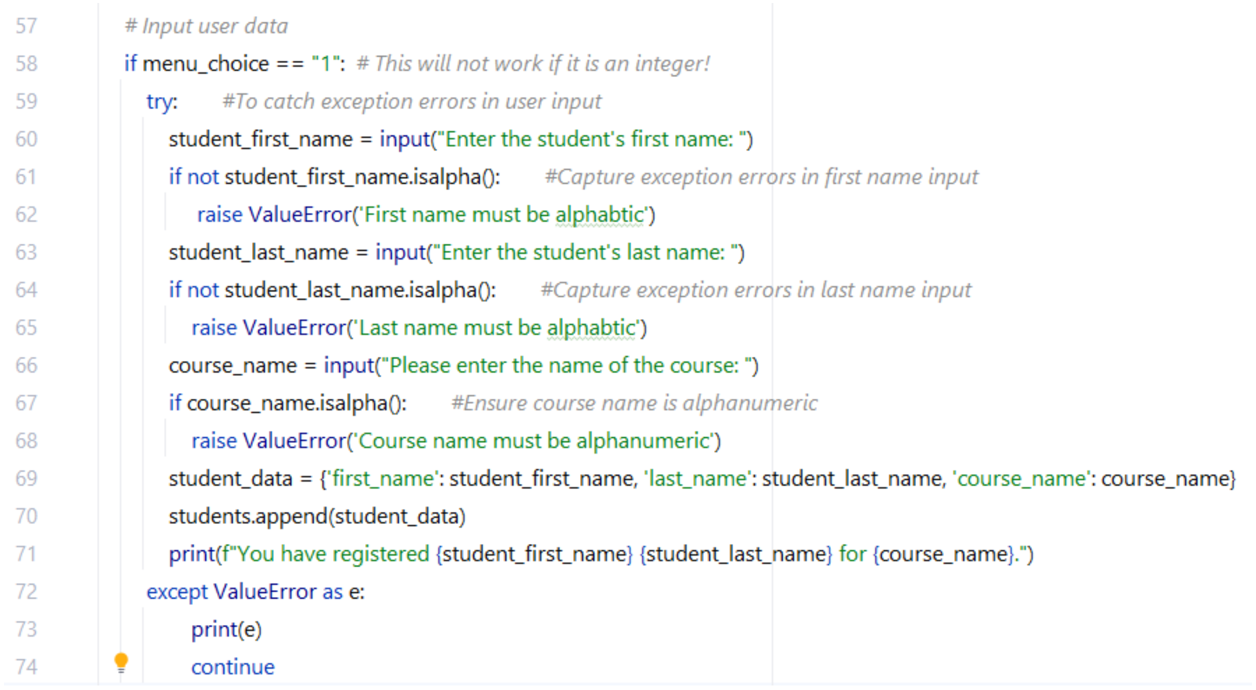
```python
# Present the current data
elif menu_choice == "2":
    # Process the data to create anc
    print("-" * 80)
    print(students, '\n')
    print("-" * 80)
    continue
```



```
---- 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
--------------------------------------------------------------------------------
[{'first_name': 'ismail', 'last_name': 'eldin', 'course_name': 'Python 101'}, {'first_name'
```

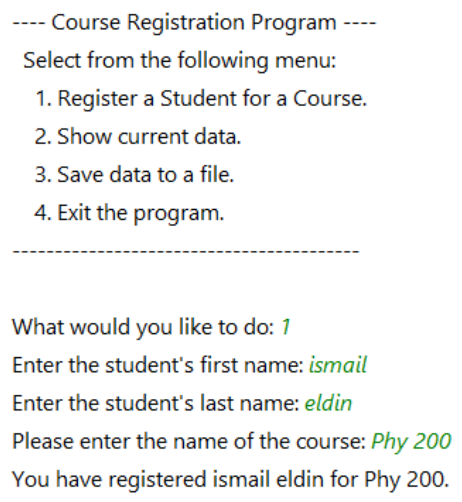
Figure 8 & 9: Menu selection 2 and output

On menu choice 1, the user is prompted to enter first and last names and course name using the input() function and the inputs are stored in the respective variables. The entered user data are appended to the students list of dictionaries but not saved. Error handling exceptions capture inappropriate entries and display messages accordingly (Figure 10).



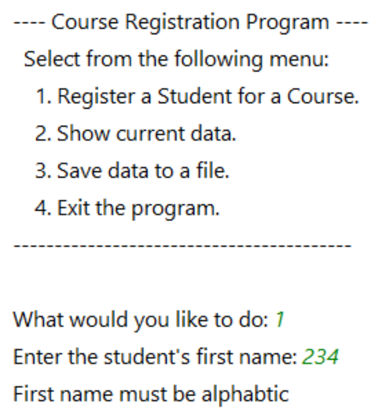
```python
# Input user data
if menu_choice == "1":  # This will not work if it is an integer!
    try:      #To catch exception errors in user input
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():      #Capture exception errors in first name input
            raise ValueError('First name must be alphabtic')
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():      #Capture exception errors in last name input
            raise ValueError('Last name must be alphabtic')
        course_name = input("Please enter the name of the course: ")
        if course_name.isalpha():      #Ensure course name is alphanumeric
            raise ValueError('Course name must be alphanumeric')
        student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': 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(e)
        continue
```

Figure 10: Selection menu 1

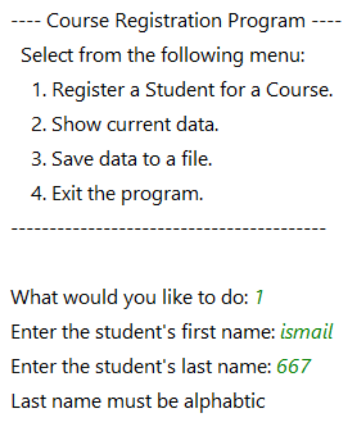


```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 1
Enter the student's first name: ismail
Enter the student's last name: eldin
Please enter the name of the course: Phy 200
You have registered ismail eldin for Phy 200.
```

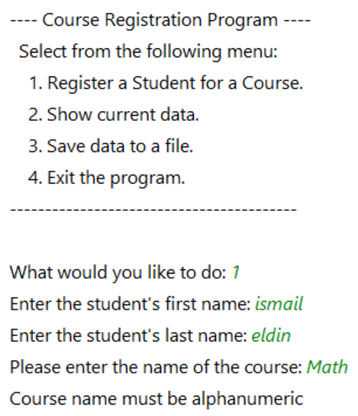
Figure 11: Output of selection menu 1



```
---- 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: 234
First name must be alphabtic
```



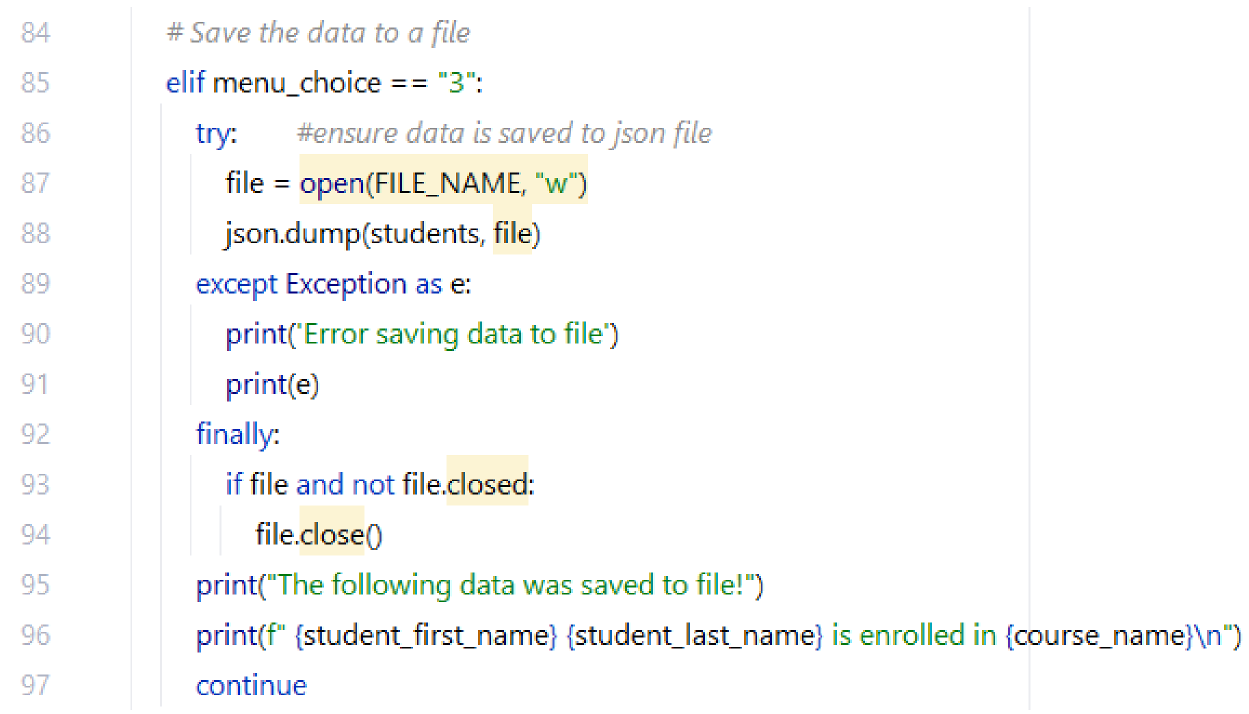
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do: 1
Enter the student's first name: ismail
Enter the student's last name: 667
Last name must be alphabtic
```



```
---- 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: ismail
Enter the student's last name: eldin
Please enter the name of the course: Math
Course name must be alphanumeric
```

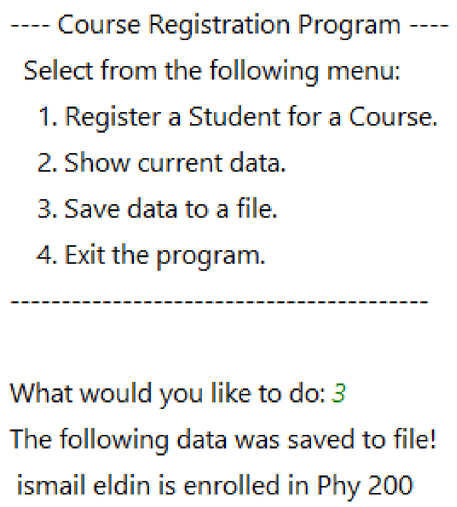
Figure 12: Error handling exceptions of user input data

Selection menu 3 opens the file Enrollments.json and writes the data with a try-except block capturing errors in file saving and finally ensuring file closure, if not closed, saving the user input data of selection menu 1 to the data already existing in the JSON file (Figure 13).



```
# Save the data to a file
elif menu_choice == "3":
    try:      #ensure data is saved to json file
        file = open(FILE_NAME, "w")
        json.dump(students, file)
    except Exception as e:
        print('Error saving data to file')
        print(e)
    finally:
        if file and not file.closed:
            file.close()
    print("The following data was saved to file!")
    print(f" {student_first_name} {student_last_name} is enrolled in {course_name}\n")
    continue
```

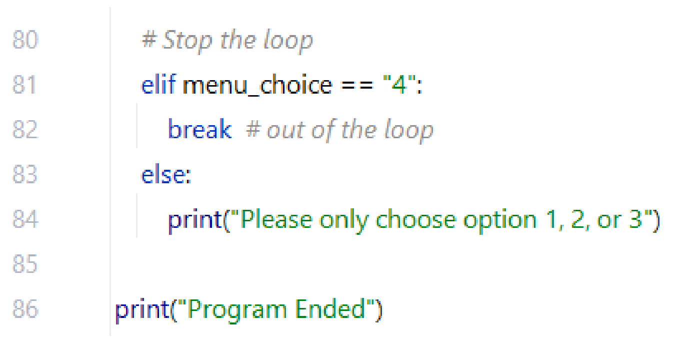
Figure 13: Selection menu 3



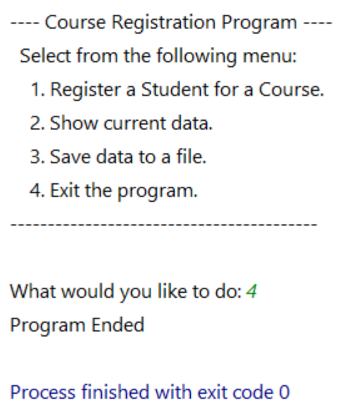
```
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
-----------------------------------------

What would you like to do: 3
The following data was saved to file!
 ismail eldin is enrolled in Phy 200
```

Figure 14: Output of selection menu 3

Selection menu 4 breaks and ends the while loop. Any other input by the user prompts the script to print “Please only choose 1, 2, or 3” (Figure 15 & 16).



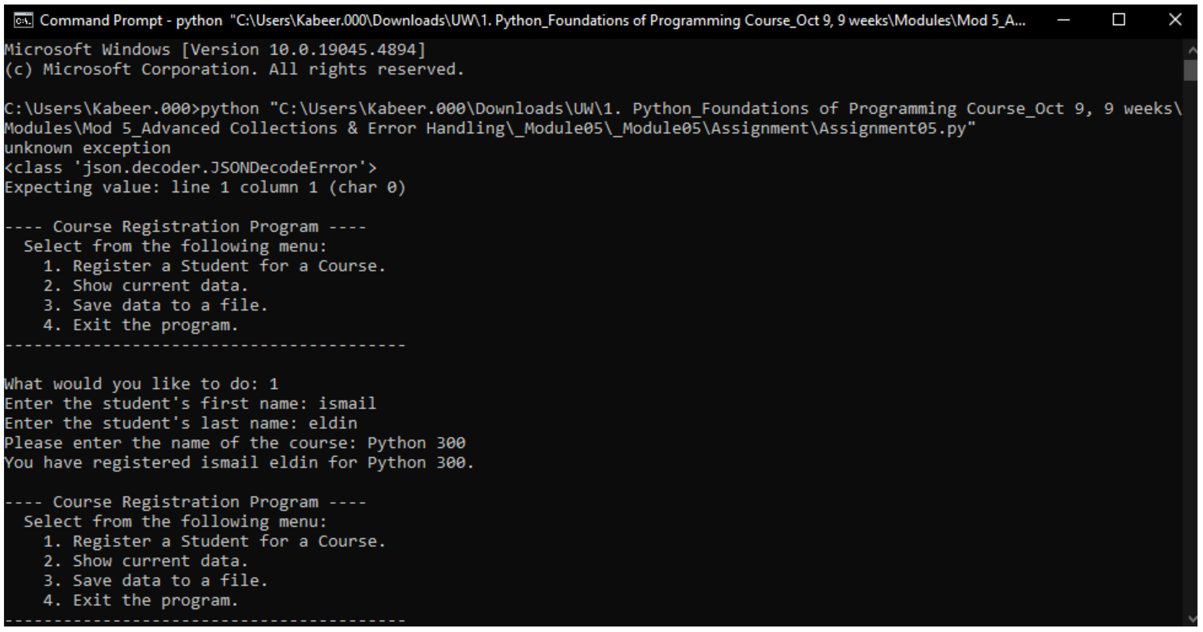
```python
    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, or 3")

print("Program Ended")
```



```
---- 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

Process finished with exit code 0
```

Figure 15 & 16: Selection menu 4 and output

The Windows Command Prompt displays the same output as the Pycharm IDLE output utilizing the Assignment05.py physical path (Figure 17).



```
Command Prompt - python "C:\Users\Kabeer.000\Downloads\UW\1. Python_Foundations of Programming Course_Oct 9, 9 weeks\Modules\Mod 5_A...
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kabeer.000>python "C:\Users\Kabeer.000\Downloads\UW\1. Python_Foundations of Programming Course_Oct 9, 9 weeks\
Modules\Mod 5_Advanced Collections & Error Handling\_Module05\_Module05\Assignment\Assignment05.py"
unknown exception
<class 'json.decoder.JSONDecodeError'>
Expecting value: line 1 column 1 (char 0)

---- 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: ismail
Enter the student's last name: eldin
Please enter the name of the course: Python 300
You have registered ismail eldin for Python 300.

---- 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 17: Windows Command Prompt Output

Summary

This short script illustrates JSON file opening, file reading, capturing user input data, appending processed data to list of dictionary rows, displaying the processed data, and then saving the processed key/value pair data as a multi-dimensional list. Error handling and exception messages capture errors in file opening, saving, input data, and ensure opening a file if one does not exist. IDLE and Windows Command Prompt outputs are presented showing identical outputs.