Ryan Miller

May 29th , 2024

IT FDN 110 B Sp 24: Foundations Of Programming: Python

Assignment 07

Creating Python Scripts

Introduction

I created a Python program that demonstrates using constants, variables, and print statements to display a menu option about a student's registration using Functions and Classes and to write to a JSON file. I named the file as assignment_07.py. I included a script header that included below text and has my name and the current date.

While working on this assignment, I learned how to –

1. How to use Functions and Classes
2. How to use static function
3. How to write to a JSON File
4. How to use json.dump command
5. How to use Git Hub
6. How to apply Error Handling
7. How to use. join() command
8. How to add data to two-dimensional list table
9. How to use FOR command
10. How to define a Menu
11. How to define choices
12. How to use WHILE command
13. How to use IF Command
14. How to use ELIF Command
15. How to Close a Loop
16. How to format strings
17. Troubleshooting and debugging my code

File Edit Format Run Options Window Help

```
#------------------------------------------------------------------#
# Title: Assignment05
# DESC: This assignment demonstrates using constants, variables,
#       operators, formatting, and files
# and calculations
# Change Log: (Who, When, What)
# Ryan, 5/28/2024, Created Script
#------------------------------------------------------------------#
```

Code

I wrote the code below with Functions and Classes. I used the While, IF and ELIF Commands. Also, I used the OPEN, WRITE AND CLOSE JSON File commands.

```
import 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
----------------------------------------
"""

FILE_NAME: str = "Enrollments.json"
menu_choice: str = ""
students: list = []

class FileProcessor:

    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        try:
            with open(file_name, 'r') as file:

                loaded_student_data = json.load(file)
                for student in loaded_student_data:
                    loaded_student = Student(first_name = student["First_Name"], last_name = student["Last_Name"], course = student["Course"])
                    student_data.append(loaded_student)

                print("File loaded")
        except FileNotFoundError as e:
            IO.output_error_messages("The text/json file could not be found when running this script", e)
        except ValueError as e:
            IO.output_error_messages("There are corruption issues in the file", e)
        except Exception as e:
            IO.output_error_messages("Unknown error.", e)
        finally:
            if file.closed == False:
                file.close()
        return student_data


    @staticmethod
    def write_data_to_file(file_name: str, student_data: list):
        try:
            list_of_students: list = []
            for student in student_data:
                student_json: dict = {"First_Name": student.first_name, "Last_Name": student.last_name, "Course": student.course}
                list_of_students.append(student_json)
```

```
    @staticmethod
    def write_data_to_file(file_name: str, student_data: list):
        try:
            list_of_students: list = []
            for student in student_data:
                student_json: dict = {"First_Name": student.first_name, "Last_Name": student.last_name, "Course": student.course}
                list_of_students.append(student_json)

            file = open(file_name, "w")
            json.dump(list_of_students, file)
            print(f"The student list was saved in {FILE_NAME}")
            file.close()
        except TypeError as e:
            IO.output_error_messages("The data may not be in a valid format", e)
        except Exception as e:
            IO.output_error_messages("Unknown error.", e)
        finally:
            if file.closed == False:
                file.close()

class IO:

    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        print(message)
        if error is not None:
            print("-- Technical Error Message --")
            print(error, error.__doc__, type(error), sep='\n')


    @staticmethod
    def output_menu(menu: str):
        print(menu)


    @staticmethod
    def input_menu_choice():
        try:
            user_choice = input("Please select an option: ")
            if user_choice not in ("1","2","3","4"):
                raise Exception("Please select only 1, 2, 3, or 4.")
        except Exception as e:
            IO.output_error_messages("Unknown Error.",e.__str__())
        return user_choice


    @staticmethod
    def output_student_courses(student_data: list):
        print("The current list of students is:")
        print("Name \t\tLast Name \tCourse")
        for student in student_data:
            print(f"{student.first_name} \t\t {student.last_name} \t\t{student.course}")
```

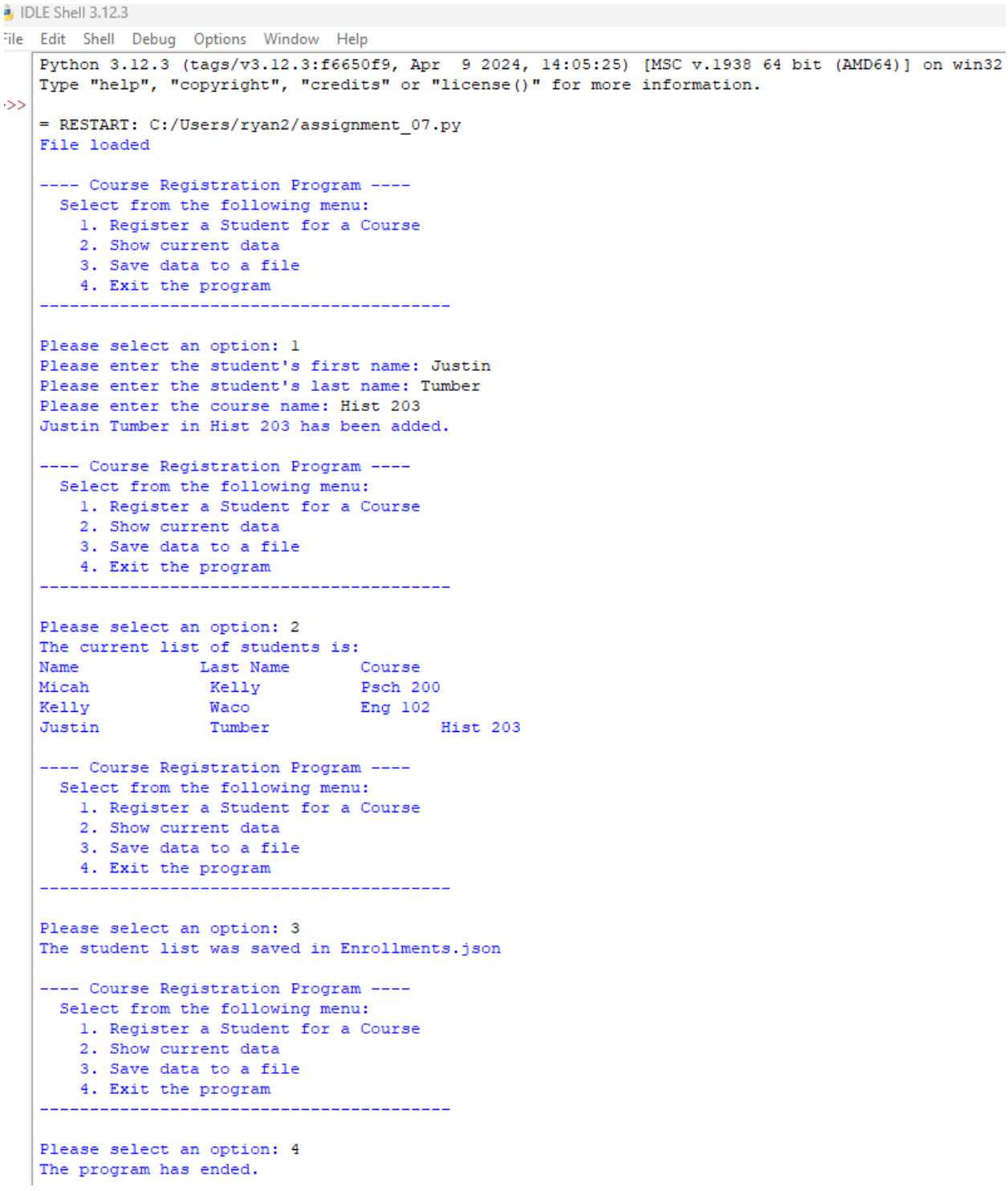
```
    @staticmethod
    def input_student_data(student_data: list):
        try:
            student_first_name = input("Please enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The first name should not contain numbers or symbols")

            student_last_name = input("Please enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The first name and last name should not contain numbers or symbols")

            course_name = input("Please enter the course name: ")

            new_student = Student(first_name=student_first_name, last_name=student_last_name, course= course_name)

            student_data.append(new_student)
            print(f"{new_student.first_name} {new_student.last_name} in {new_student.course} has been added.")
        except ValueError as e:
            IO.output_error_messages("There is a incorrect value.", e)
        except Exception as e:
            IO.output_error_messages("Unknown error.", e)
        return student_data

class Person:

    def __init__(self, first_name: str = "", last_name: str = ""):
        self.first_name = first_name
        self.last_name = last_name

    @property
    def first_name(self):
        return self.__first_name.title()

    @first_name.setter
    def first_name(self, value: str):
        if value.isalpha() or value == "":
            self.__first_name = value
        else:
            raise ValueError("There are numbers or symbols in the first name!")

    @property
    def last_name(self):
        return self.__last_name.title()

    @last_name.setter
    def last_name(self, value: str):
        if value.isalpha() or value == "":
            self.__last_name = value
        else:
            raise ValueError("There are numbers or symbols in the first name!")

    def __str__(self):
        return f"{self.first_name},{self.last_name}"

class Student(Person):

    def __init__(self, first_name: str, last_name: str, course: str):
        super().__init__(first_name = first_name, last_name= last_name)
        self.course = course
```

```
class Student(Person):

    def __init__(self, first_name: str, last_name: str, course: str):
        super().__init__(first_name = first_name, last_name= last_name)
        self.course = course

    @property
    def course(self):
        return self.__course

    @course.setter
    def course(self, value: str):
        self.__course = value

    def __str__(self):
        return f"{self.first_name},{self.last_name},{self.course}"

if __name__ == "__main__":

    students = FileProcessor.read_data_from_file(file_name = FILE_NAME, student_data = students)

    while True:
        IO.output_menu(MENU)

        menu_choice = IO.input_menu_choice()

        match menu_choice:
            case "1":
                students = IO.input_student_data(student_data = students)
            case "2":
                IO.output_student_courses(student_data = students)
            case "3":
                FileProcessor.write_data_to_file(file_name = FILE_NAME, student_data = students)
            case "4":
                print("The program has ended.")
                exit()
```

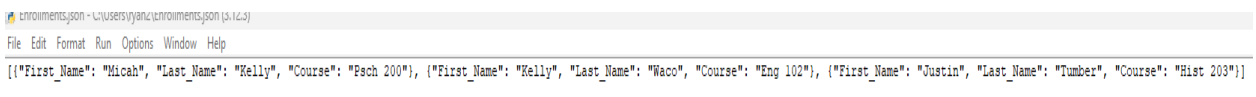
Output

Below is a depiction of my code output in IDLE Shell, JSON File and Command Prompt.

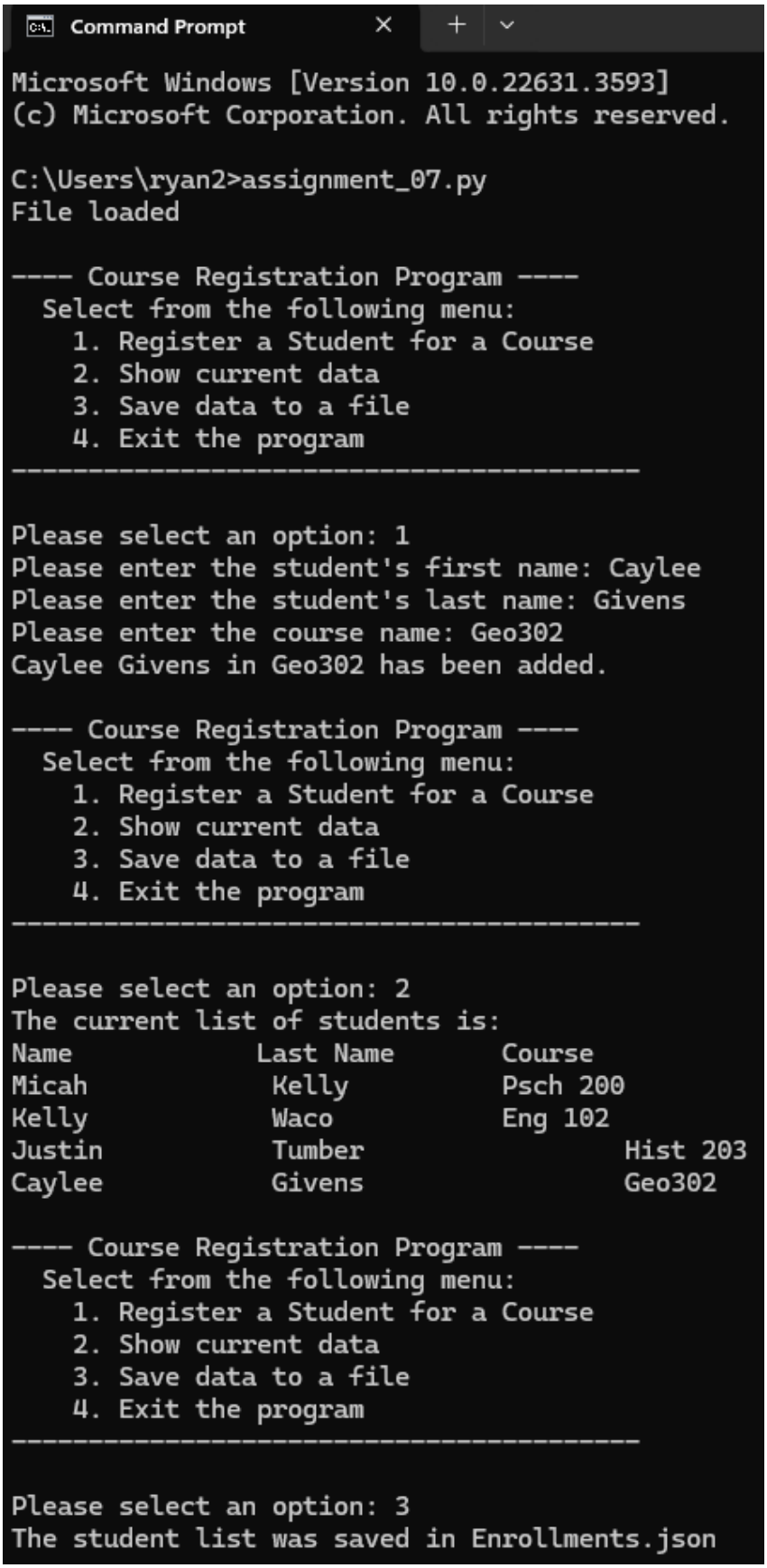
1. **IDLE Shell**

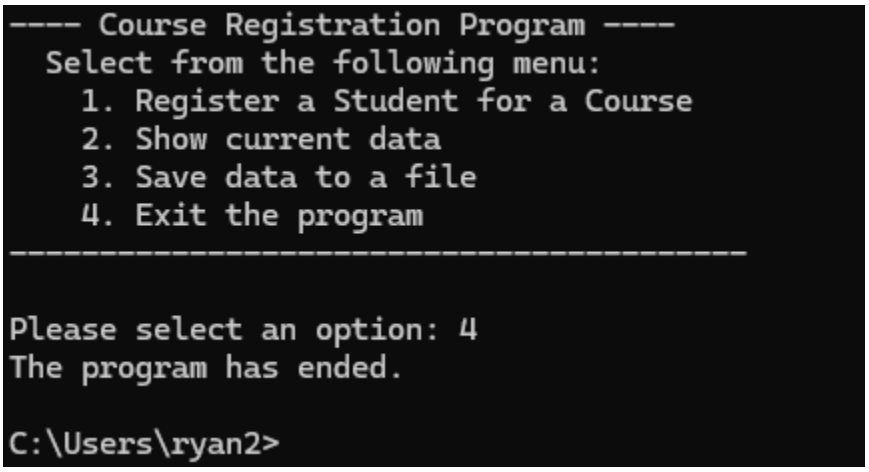

IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>
= RESTART: C:/Users/ryan2/assignment_07.py
File loaded
---- Course Registration 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
Please select an option: 1
Please enter the student's first name: Justin
Please enter the student's last name: Tumber
Please enter the course name: Hist 203
Justin Tumber in Hist 203 has been added.
Please select an option: 2
The current list of students is:
Name Last Name Course
Micah Kelly Psch 200
Kelly Waco Eng 102
Justin Tumber Hist 203
Please select an option: 3
The student list was saved in Enrollments.json
Please select an option: 4
The program has ended.

2. JSON File


File Edit Format Run Options Window Help
[{"First_Name": "Micah", "Last_Name": "Kelly", "Course": "Psch 200"}, {"First_Name": "Kelly", "Last_Name": "Waco", "Course": "Eng 102"}, {"First_Name": "Justin", "Last_Name": "Tumber", "Course": "Hist 203"}]

3. **Command Prompt**


Command Prompt
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.
C:\Users\ryan2>assignment_07.py
File loaded
---- 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
Please select an option: 1
Please enter the student's first name: Caylee
Please enter the student's last name: Givens
Please enter the course name: Geo302
Caylee Givens in Geo302 has been added.
---- 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
Please select an option: 2
The current list of students is:
Name Last Name Course
Micah Kelly Psch 200
Kelly Waco Eng 102
Justin Tumber Hist 203
Caylee Givens Geo302
---- 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
Please select an option: 3
The student list was saved in Enrollments.json



Summary

For being able to write above code and successfully run it, I had watched all the assigned videos and reading. Also, watching the class recording helped me as well. While I was writing the code, I initially had a lot of bugs, so I had go back and rewatch most of the videos to clarify some of my questions around adding a Menu Choice and writing to a JSON File.