Ace Bailey
July 31, 2024
IT FDN 110 A
Assignment 05

Python Course Registration Program

Intro

This week, I built more on my code. I changed the data processing to use dictionaries and exception handling. We learned this week about JSON Files, Try/Except, and Github.

Creating the Program

The program started with information about the program such as the title and description. I then included a change log to better monitor when I changed the program and how. I then created my constants with unchanging data such as the file name and menu string. I used import json to import the json module to later use with JSON data.

```
#
------------------------------------------------------------------------------
----------- #
# Title: Assignment05
# Desc: Data processing using dictionaries and exception handling
# Change Log: (Who, When, What)
#   RRoot,1/1/2030,Created Script
#   Aisling,7/31/2024,Updated Script to meet acceptance criteria
#
------------------------------------------------------------------------------
----------- #

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.json"
```

After defining the constants, I defined the variables. I defined student_data as an empty dictionary that will later hold student data. I can change the dictionary to store key-value pairs. Students is defined as an empty list and json_data is an empty string.

```
# Define the Data Variables
student_first_name: str = ''
student_last_name: str = ''
course_name: str = ''
student_data: dict = {}
students: list = []
json_data: str = ''
file = None
menu_choice: str = ''
```

Next I use the try block to open the file and load the contents into students. This will return an error using except if the file does not exist or if there is an error reading it and then prints an error message.

```
# When the program starts, the contents of json are read into a two-dimensional
list table
try:
   with open(FILE_NAME, "r") as file:
       students = json.load(file)
except FileNotFoundError:
   print(f"{FILE_NAME} not found. Starting with an empty list.")
except json.JSONDecodeError:
   print(f"Error reading {FILE_NAME}. Starting with an empty list.")
except Exception as e:
   print(f"An error occurred: {e}")
```

I then made the main while loop. This will read the choice from 1-4 that the user inputs and move onto the if loop below it as long as that choice is not 4 which will end the program. Data entered is stored in the menu_choice variable after the MENU constant is printed. If the menu_choice is equal to 1 it will register a student for a course by taking in the input of the students information into variables which will then be put into student_data and added to students using ".append". The information will then be read back to the user. In between the inputs taken from the user are error handling to make sure the first and last names are not empty. If they are empty "ValueError" is used to allow the user to reinput something.

```
# Process and read back to user the data
while True:
   print(MENU)
   menu_choice = input("What would you like to do: ")
```

```
    if menu_choice == "1":
        # Register a student for a course
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name:
                raise ValueError("First name cannot be empty.")
        except ValueError as e:
            print(e)
            continue

        try:
            student_last_name = input("Enter the student's last name: ")
            if not student_last_name:
                raise ValueError("Last name cannot be empty.")
        except ValueError as e:
            print(e)
            continue

        course_name = input("Please enter the name of the course: ")
        student_data = {"first_name": student_first_name, "last_name":
student_last_name, "course": course_name}
        students.append(student_data)
        print(f"You have registered {student_first_name} {student_last_name} for
{course_name}.")
        continue
```

If the user did not choose 1, the code will continue to menu_choice 2. If that is what the user input then it will show the current data stored in the students list. It does this by printing each student data in students using a for loop.

```
elif menu_choice == "2":
    # Show current data
    print("-" * 50)
    for student in students:
        print(f"Student {student['first_name']} {student['last_name']} is
enrolled in {student['course']}")
    print("-" * 50)
    continue
```

If the user input 3 then the data will be saved to a json file. Data will be saved to Enrollments.json using the try block. It opens the file and writes the content using ".dump". It will then print a confirmation message. There is error handling that uses "Exeption" to print an error function if there was an error in saving the file.

```
elif menu_choice == "3":
   # Save the data to a file
   try:
       with open(FILE_NAME, "w") as file:
           # Write content into file
           json.dump(students, file)
       print("Your data was saved")
       for student in students:
           print(f"Student {student['first_name']} {student['last_name']} is
enrolled in {student['course']}")
   except Exception as e:
       print(f"An error occurred while saving the file: {e}")
   continue
```

If the user chooses 4 then the code will exit. The break statement terminates the while loop. Lastly, if none of the 4 options are chosen, a statement will be printed asking the user to pick a viable option. The program will end and let the user know.

```
elif menu_choice == "4":
   # Ends code
   break
else:
   print("Please choose a viable option")

print("Program Ended")
```

Testing the Program

To test that the program worked, I first saved the script as Assignment05.py. After saving the script I ran it in Pycharm. It returned the desired output as seen in figure 1 and 2.

```
---- 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: Ace
Enter the student's last name: Bailey
Please enter the name of the course: Python100
You have registered Ace Bailey for 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
------------------------------------------------
Student ace bailey is enrolled in p100
Student Ace Bailey is enrolled in Python100
------------------------------------------------
```

Figure 1: Output in Pycharm (part 1)

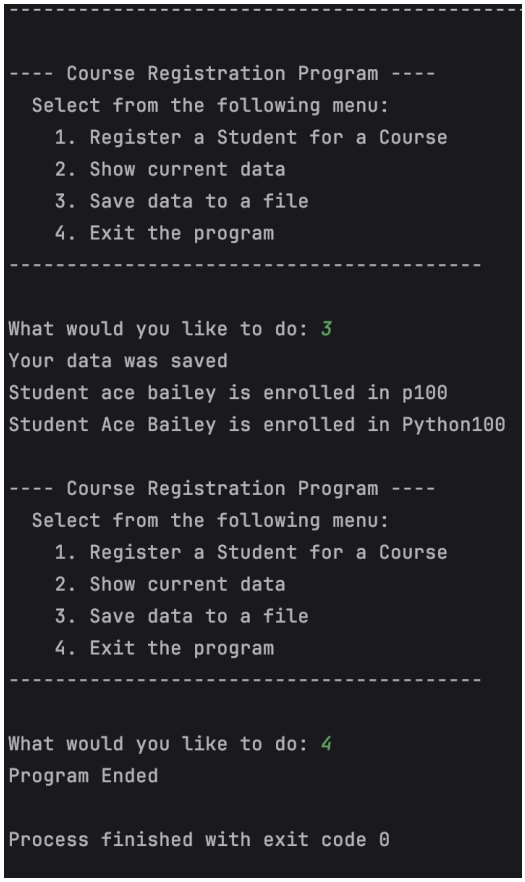

Figure 2: Output in Pycharm (part 2)

I then ran the code in Terminal which also went well.

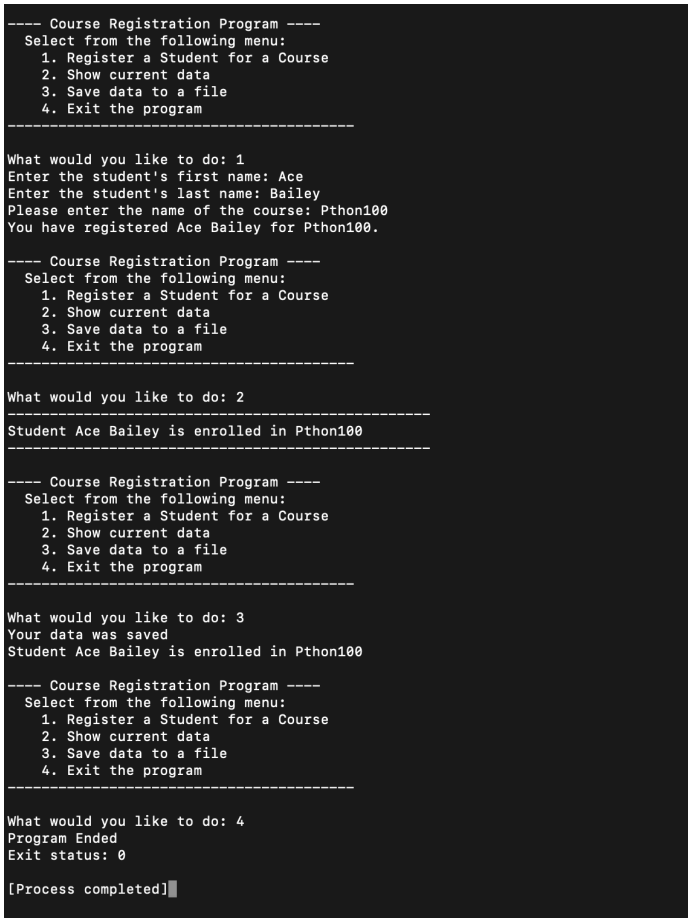

Figure 3: Output in Terminal

Summary

The program that I wrote this week shows how dictionaries can be easily used to store information and exception handling can be used to allow users to correct errors.