Ivan Esley

12th February 2024

IT FDN 110

Assignment 05

Advanced Collections and Error Handling

Introduction

For this assignment, the task we were given was to create python program that demonstrates **adding the use of data processing using dictionaries and error handling.** This paper is a documentation on how I completed this task and my understanding for this week's task.

The Script

assignment05_ivanesley.py > ...

```python
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   RRoot,1/1/2030,Created Script
#   Ivan Esley, 2/12/24, Final edits/submission
# ------------------------------------------------------------------------------------------ #

import csv

# 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.csv"

# 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 = {}  # setting student_data as dictionary
students: list = []  # a table of student data
csv_data: str = ''  # Holds combined string data separated by a comma.
file = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.

```

Figure 1. Defining data constants, variables and menu options

We define the data constants MENU & FILE_NAME as shown above. It is also defined as a string. These constants are set so they shouldn't be changed. Figure 1 shows how all the variables are defined. This is set to change according to what the user inputs. Here it has also

been defined beforehand if it will be a string or a float. We also defined our dictionary variable as seen on line 29.

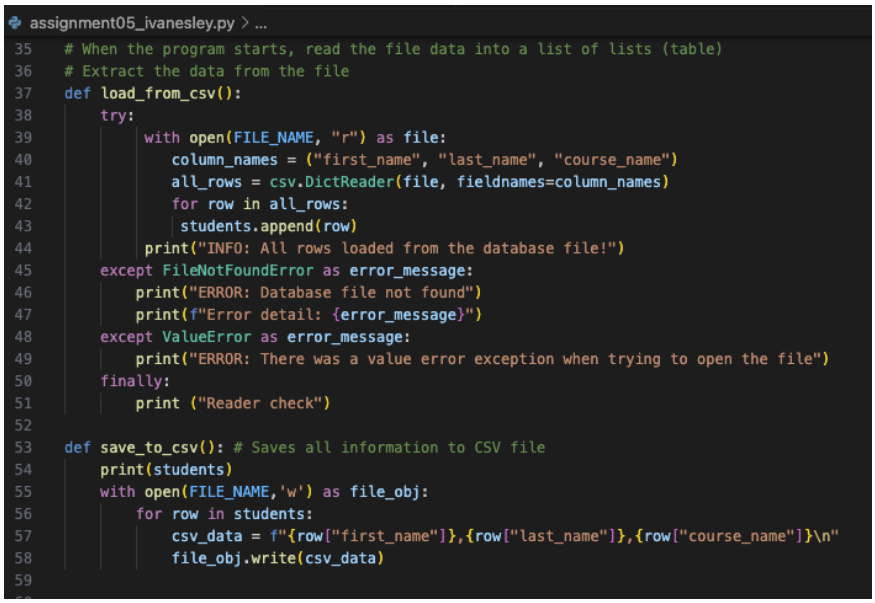

Figure 2. Main script/ processing different menu options pt1

Line 37 and 53 is just defining the two variables that will allow the code to save the data to a csv file. The first one is a read file that creates it and the next other one is a write file which inputs the data entered by the user into the file. The use of the dictionary reader and writer can be seen here. If the cvs reader can't run it runs through the error messages which will tell the user exactly what is wrong this is an example of the error handling.

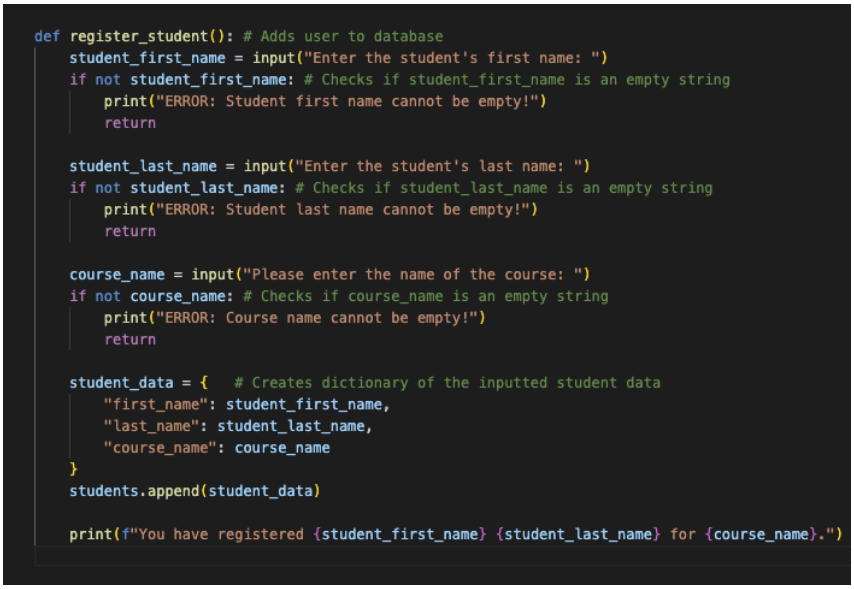

Figure 3. Main script/ processing different menu options pt2

This section as seen on figure 3, asks the user to input their first name, last name and course name. If the user doesn't enter anything in the first & last name section is will print an error message and take the user back to the menu. We can also see that student_data is being used as a dictionary here for all the data that the user is inputting. The append() is a used which allows the user to adds (an) additional element(s) to the end of the selected parent element (students).

```
# 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!
       register_student()

    # 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["first_name"]} {student["last_name"]} is enrolled in {student["course_name"]}")
        print("-"*50)
        continue

 # Save the data to a file
    elif menu_choice == "3":
        save_to_csv()

 # 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")
```

Figure 4. Main script/ processing different menu options pt 3

The first print function and menu_choice string prints the MENU options to the user with a text that prompts them to choose from the options displayed. The "while True" statement prints the MENU again to the user once they chose an option.

The if statement lets the user chose the option 1 from the MENU. The code then refers the variable register_student which was define earlier.

The "elif statement" activates if option 2 is picked, it presents a coma-separated string by formatting the collected data using the print function.

When option 3 from the menu is picked the program opens a file named "Enrollments.csv" in write mode using the open() function.

Lastly when option 4 is picked the program ends. If an invalid option is chosen, we get a print statement prompting us to pick a valid option.

Testing the Script

-The program takes the user's input for a student's first, last name, and course name.
-The program displays the user's input for a student's first, last name, and course name.
-The program saves the user's input for a student's first, last name, and course name to a coma-separated string file.
-The program allows users to enter multiple registrations (first name, last name, course name).
-The program allows users to display multiple registrations (first name, last name, course name).
-The program allows users to save multiple registrations to a file (first name, last name, course name).
-The program runs correctly in both PyCharm and from the console or terminal.

Summary

In conclusion, the task was achieved. The script ran on visual studio IDE, terminal and IDLE like it should have. This task helped me learn how to use data processing using dictionaries and error handling