Name: Gustavo Limongi Araujo

Date: 02/13/2024

Course: Introduction to Python

Assignment 05

Intro

Create a Python program that demonstrates using constants, variables, and print statements to display a message about a student's registration for a Python course. This program is very similar to Assignment04, but **It adds the use of data processing using dictionaries and exception handling.**

Topic

Using PyCharm as an IDE, the header is the following:

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Gustavo Limongi Araujo, 02/13/2024, Assignment05)
#   RRoot,1/1/2030,Created Script
#   <Gustavo L. Araujo>,<2/13/2024>, <Assignment 05>
# ------------------------------------------------------------------------------------------ #
```

The menu choice and the file name were defined below the header:
The variables were defined as stringer: student first and last name, course name, the csv data and file obj.

The FILE_NAME was also defined as a stringer

All empty initially:

```
# Define the Constants
MENU = """
-------------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
-----------------------------------------------------
"""
```

Then to make the code cleaner I defined 3 functions using def () to register the student, print data and save the data file to csv file using dictionaries with key and value.

1) Function **register_student**: read data from student using the *input* function, such as first name, last name and course name. It appends the data in a dictionary to the *student_data.*
2) Function **show_current_data**: print what was stored in the variable csv_data using *print* function. It iterates over each dictionary in the student_data list and prints the first, last name and course of each student. The print used f-string formatting to insert the values from the dictionary.

3) Function **save_data_to_file**: this function saves data stored in student_data list to a CSV file specified in the FILE_NAME. The with open, opens the file specified in the FILE_NAME in write mode. The newline= '' prevent extra blank lines being inserted between rows in the CSV file. The headers define the headers in the csv file. The csv_writer = csv.dictwriter creates a CSV writer object using DictWriter class from the csv module, specifying the fieldnames (header) for the CSV file. The csv_writer.writeheader() writes the headers to the CSV file.
   The for loop iterates over each dictionary in the student_data list. The csv_writer.writerow for each student, writes a row to the CSV file using the student's data (dictionary)
   The print with f-string prints a message indicating the data was successfully saved.
   The error exception was used if any exception occurs during the saving process, this catches it and prints an error message.

```
def register_student():
    global student_first_name, student_last_name, course_name
    student_first_name = input("Student First Name: ")
    student_last_name = input("Student Last Name: ")
    course_name = input("Course name: ")
    student_data.append({"First Name": student_first_name, "Last Name": student_last_name, "Course": course_name})

def show_current_data():
    for student in student_data:
        print(f"First Name: {student['First Name']}, Last Name: {student['Last Name']}, Course: {student['Course']}")

def save_data_to_file():
    try:
        with open(FILE_NAME, 'w', newline='') as file:
            headers = ["First Name", "Last Name", "Course"]
            csv_writer = csv.DictWriter(file, fieldnames=headers)
            csv_writer.writeheader()
            for student in student_data:
                csv_writer.writerow(student)
        print(f"Data saved to {FILE_NAME}")
    except Exception as e:
        print(f"Error saving data to a file: {e}")
```

The function **read_csv_data()** is called in the main program and reads the data from CSV file into a list of dictionaries. It opens the file specified by FILE_NAME in read mode. If the file is found, it creates a CSV object csv_reader using DictReader. It iterates over each row in the CSV file, appending each row (as dictionary) to the student_list. If the file is not found, it prints a message.

If any other exceptions occurs during the reading process, it prints an error message along with the specific exception that occurred.

```
def read_csv_data():
    try:
        with open(FILE_NAME, 'r') as file:
            csv_reader = csv.DictReader (file)
            for row in csv_reader:
                students.append(row)
    except FileNotFoundError:
         print(f"File {FILE_NAME} not found")
    except Exception as e:
         print(f"Error reading file :{e}")
```

Finally, I used a while function that repeats until the menu choice is True. If it is 4, it exists the program. Then a simple *if* and *elif* commands were used to call each function above, hence:

If 1: calls function *register_account*

If 2: calls function *print_data*

If 3: *save_data_to_file*

If 4: exits the program

```
# Main Program
read_csv_data()

while True:
    print(MENU)
    menu_choice = input("Enter your choice from (1-4): ")

    if menu_choice == "1":
        register_student()
    elif menu_choice == "2":
        show_current_data()
    elif menu_choice == "3":
        save_data_to_file()
    elif menu_choice == "4":
        print("Exiting the program....")
        break
    else:
        print("Invalid Choice. Please try again")
```

Summary

The videos and course notes of Lesson 04 were used to help write this code.

The code was tested with PyCharm and gitbash and worked as intended.