Creating Python Scripts

Name: Priya Tomar
Date: 08/28/2024
Course: IT FDN 110 B Su 24: Foundations of Programming: Python

Introduction

This document outlines the steps and considerations involved in developing the Python script for managing student course registrations, as specified in Assignment05. The assignment demonstrates the use of lists and files for data processing and includes functionalities for registering students, displaying data, and saving data to a CSV file. The goal was to create a functional program that meets the given specifications while incorporating error handling.

Topic Covered:

Steps Taken to Perform the Assignment

Step 1: Understanding the Requirements

I began by thoroughly reviewing the assignment's instructions to understand the program's requirements. The task was to create a Python program with the following functionalities:

- Register students for a course.
- View the entered data.
- Save the data to a CSV file.
- Exit the program.

This step involved identifying the core requirements and constraints for the program, ensuring that each feature would be addressed in the final implementation.

Step 2: Setting Up the Script

I created a Python script named Assignment05.py and included a header to document the script's purpose, description, and change log. This header helps in tracking updates and understanding the script's functionality.

Figure 1: Header

```
1  # ------------------------------------------------------------------------------------------ #
2  # Title: Assignment05.py
3  # Desc: This assignment demonstrates managing student course registrations with options to regis
4  # Change Log: (Who, When, What)
5  #  Priya, 8/28/2024, Updated Script with additional error handling and functionalities
6  # ------------------------------------------------------------------------------------------ #
```

Step 3: Setting up constants and variables

I defined constants and variables used throughout the program:

Figure 2: Constants and variables

```
# 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.csv"

# Define the Data Variables
student_first_name: str = ""
student_last_name: str = ""
course_name: str = ""
csv_data: str = ""
file = None
menu_choice: str = ""
student_data: list = []
students: list[list[str]] = []
```

Step 4: Building the main program loop:

I implemented a while loop to create a main menu that continuously prompts the user until they choose to exit. The menu provides options to:

- Register a student.
- Display current data.
- Save data to a file.
- Exit the program.

Figure 3: Main program loop

```
try:
    with open(FILE_NAME, 'r') as file_obj:
        for row in file_obj:
            parts = row.strip().split(',')
            if len(parts) == 3:
                students.append(parts)
except FileNotFoundError:
    print(f"File {FILE_NAME} not found. Starting with an empty list.")
except Exception as e:
    print(f"An error occurred while reading the file: {e}")

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

```
    # Input user data
    if menu_choice == "1":
        try:
            student_first_name = input("Enter student's first name: ").strip()
            if not student_first_name:
                raise ValueError("First name cannot be empty.")
            student_last_name = input("Enter student's last name: ").strip()
            if not student_last_name:
                raise ValueError("Last name cannot be empty.")
            course_name = input("Enter course name: ").strip()
            if not course_name:
                raise ValueError("Course name cannot be empty.")

            student_data = [student_first_name, student_last_name, course_name]
            students.append(student_data)
            print(f"You have registered {student_first_name} {student_last_name} for {course_na

        except ValueError as ve:
            print(f"Input error: {ve}")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")
```

```
    # Present the current data
    elif menu_choice == "2":
        print("\nThe current data is:")
        if students:
            for student in students:
                print(f"{student[0]}, {student[1]} is enrolled in {student[2]}")
        else:
            print("No data to display.")
        print()

    # Save the data to a file
    elif menu_choice == "3":
        try:
            with open(FILE_NAME, 'w') as file_obj:
                for student in students:
                    file_obj.write(f"{student[0]},{student[1]},{student[2]}\n")
            print(f"Data has been saved to {FILE_NAME}.")
        except IOError as io_err:
            print(f"File error: {io_err}")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")

    # Exit the program
    elif menu_choice == "4":
        print("Exiting the program.")
        break

    else:
        print("Please only choose option 1, 2, 3, or 4.")
```

Step 5: Testing the program

I thoroughly tested the program using PyCharm to ensure all functionalities were working as expected. During testing, I verified:

- The program correctly handled user inputs.
- Data was displayed accurately.
- Data was saved to the CSV file properly.

Output:

The script successfully collected user input, formatted the data, and saved it to a CSV file

Figure 7: Output

```
D:\Python_Charm_Projects\assignement_04\.venv\Scripts\python

---- 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 student's first name: priya
Enter student's last name: tomar
Enter course name: Python
You have registered priya tomar for Python

---- 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 student's first name: Priyanka
Enter student's last name: Sagar
```

```
Enter course name: Pythin
You have registered Priyanka Sagar for Pythin

---- 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

The current data is:
Priya, Tomar is enrolled in Python 100
sw, sw is enrolled in de
df, fg is enrolled in ww
priya, tomar is enrolled in Python
Priyanka, Sagar is enrolled in Pythin


---- 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
```

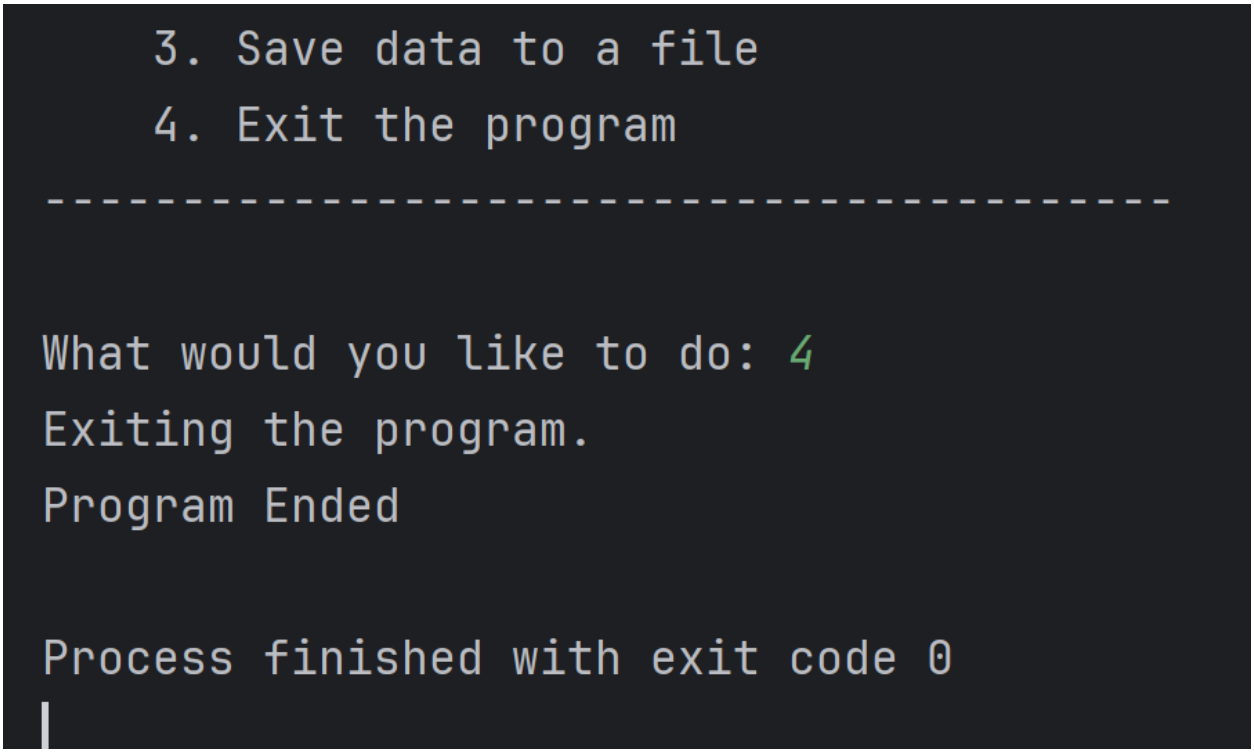
```
What would you like to do: 2

The current data is:
Priya, Tomar is enrolled in Python 100
sw, sw is enrolled in de
df, fg is enrolled in ww
priya, tomar is enrolled in Python
Priyanka, Sagar is enrolled in Pythin


---- 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
Data has been saved to Enrollments.csv.

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course
    2. Show current data
```



Summary

This assignment was a valuable exercise in applying fundamental Python programming concepts to develop a functional course registration program. By following the assignment instructions and leveraging provided resources, I successfully created a program that meets the required specifications. The process of building and testing the program enhanced my understanding of Python's capabilities, particularly in terms of user interaction and file operations. The experience reinforced key programming skills and demonstrated the importance of robust error handling in software development.