

Jesse Huminski

10/27/2024

IT FDN 100 A

Assignment 0

Enrolling A Student in A Class

Introduction

In module 03, we learned about pseudo-code, a way to outline the type of program you wish to create without having to utilize all the technical jargon within the programming language. Additionally, we learned how to begin utilizing Integrated Development Environments (IDEs) like PyCharm which will help us further develop programs and allow us to use their built-in tools without having to use a bunch of other software options. This module also taught us how to control the flow of the program, it expanded on the concept of “Conditional Statements,” as showed us how to use loops within our program.

At the end of this module, we were tasked with creating a program that will gather information from a student to register them with a specific course. This program utilizes “match-and-case” statements, “while” loops, and uses what we learned in the previous module to write the data gathered to a CSV file.

Creating the Program

As always, we start with a script header to describe the code we are writing and the changes made to the script. (Image 1.1)

```
1  # ----- #
2  # Title: Assignment03
3  # Desc: This assignment demonstrates using conditional logic and looping
4  # Change Log: (Who, When, What)
5  #   J. Huminski, 10/27/2024, Created Script
6  # ----- #
```

Image 1.1: Script Header

I define the constants that are going to be used in this script and create a menu of options for the user to choose from. (Image 1.2)

```
8      # Define the Data Constants
9      MENU: str = ("----Course Registration Program----\n\n"
10                  "Select from the following menu:\n\n"
11                  "1. Register a student for the course\n"
12                  "2. Show current data\n"
13                  "3. Sava data to a file\n"
14                  "4. Exit the program"
15                  "\n-----\n")
16      FILE_NAME: str = "Enrollments.csv"
```

Image 1.2: Defining the constants of the program.

Moving on, it's time to define the data variables the program is going to be using later down the line. All of these strings are set to "None" because we will be defining their data as we build the code further. (Image 1.3)

```
18      # Define the Data Variables
19      student_first_name: str = ""
20      student_last_name: str = ""
21      course_name: str = ""
22      csv_data: str = ""
23      file_obj = None
24      menu_choice: str = ""
```

Image 1.3: The data variables

Next, it's time to begin where we want the program to start looping. I have chosen to start the loop before the menu is printed to the user. This will print the menu for the user again after they have successfully completed an option which will allow the user to view the available options without having to scroll to the beginning of the program. (Image 1.4)

```
# Present the menu of choices
while True:
    print(MENU) # This prints the menu of options to display to the user.
```

Image 1.4: Begin the loop and print the menu of options for the user

After having the menu of options displayed to the user, the user is prompted to choose an option from the menu to work through. This is where we start to use the “Match and Case” function which will run through a specific block of code depending on the option the user chooses. The first option gathers information from the user, such as their first and last name and which class they are wanting to sign up for. (Image 1.5)

```
30 # Input user data
31 menu_choice = input("Please enter a menu option from the list above: ")
32 match menu_choice:
33     case "1":
34         print("Please enter the following information to "
35               "register a new student.\n") # Instructs user what info is needed
36         student_first_name = input("Please enter the student's first name: ") # Requests first name
37         student_last_name = input("Please enter the student's last name: ") # Requests last name
38         course_name = input("Please enter the course name you wish to register for: ") # Requests course name
39
40         print(f"\nCongratulations, {student_first_name} {student_last_name} "
41               f"has successfully registered for {course_name}!\n") # Success message
42         continue
```

Image 1.5: Gathering information from the user.

After this section is completed, the “Continue” statement brings us back to where the menu is printed and prompts the user again to select an option of what they wish to do next. Option 2 will display the data collected from the user in a message formatted for a CSV file. Through the use of “If-Else” statements, I went one step further here to ensure that if the user has not entered their information yet (i.e., the variables are undefined), it will inform them that they need to go back to select option 1. (Image 1.6)

```
43 # Present the current data
44     case "2":
45         if not student_first_name or not student_last_name or not course_name: # Check if variables hold data.
46             print("You have not input valid data. "
47                   "Please select Option 1 to input your data.\n") # Print message to go back and enter data.
48         else:
49             csv_data = f"{student_first_name},{student_last_name},{course_name}\n"
50             print(csv_data) # Prints the formatted CSV data
51             continue
```

Image 1.6: Presenting the collected information as in a CSV format.

Now, it’s time to allow the program to save the data to a CSV file. Again, I used the code from the previous section to check and ensure that the variables have been assigned appropriately. If the user chooses this option without first going through Option 1, they will be told to go back and try again. This section of the code uses the open(), write(), and close() functions that we learned in module 2. If successful, a message will be displayed letting the user know their data has been saved. (Image 1.7)

```

53 # Save the data to a file
54     case "3":
55         if not student_first_name or not student_last_name or not course_name: # Check if variables hold data.
56             print("You have not input valid data. ")
57             "Please select Option 1 to input your data.\n") # Print message to go back and enter data
58         else:
59             file_obj = open(FILE_NAME, "w") # Opens the constant FILE_NAME
60             file_obj.write(csv_data) # Writes the CSV data to the file
61             file_obj.close() # Closes and saves the CSV file
62             print("\nYour request has been processed and saved.\n") # Success message!
63             continue

```

Image 1.7: Saving the data to a CSV file.

Finally, we needed a way to exit the loop properly. If the user chooses option 4, this will execute the “break” function, display a message to the user, and exit the program. (Image 1.8)

```

65 # Stop the loop
66     case "4":
67         print("\nYou have exited the program") # Informs the user they've exited the program.
68         break

```

Image 1.8: Using the “break” function.

Though we did not learn how to do this, I ran into an issue where, when prompted to choose an option, I would type my name instead and press enter. I wasn’t paying attention and every time I would press enter, I was wondering why the program wasn’t continuing as expected. So, I figured out how to print out an error message should the user enter an invalid option. (Image 1.9)

```

70 # Correcting the user option:
71     case _: # Ensures user is entering a proper menu option.
72         print("\nInvalid option. Please read the menu "
73             "and choose an applicable option.\n") # Informs user their option is wrong.
74         continue

```

Image 1.9: Prints an error message if the user enters anything outside of 1-4.

Testing the program

Now, it’s time to run the program to ensure it is working appropriately. Using PyCharm, I am going to use the keyboard shortcut “Shift+F10” to run the program. (Image 2.1)

```
"C:\Users\jesse\Documents\Python Course\Python Scripts\pythonProject\.venv\Scripts\python.exe" "C:\Users\jesse\Documents\Python Course\Module 3\A03\Assignment03.py"
---Course Registration Program---

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above:
```

Image 2.1: Program starts as expected and prompts the user to choose an option.

First, I want to test to ensure that options 2 and 3 will prompt the user to first enter their information. Here, I will enter option 2. (Image 2.2)

```
Please enter a menu option from the list above: 2
You have not input valid data. Please select Option 1 to input your data.

---Course Registration Program---

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above:
```

Image 2.2: An error message is displayed because variable data has not been received from the user.

This is working as intended. Now, let's try option 3. (Image 2.3)

```
Please enter a menu option from the list above: 3
You have not input valid data. Please select Option 1 to input your data.

----Course Registration Program----

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above: |
```

Image 2.3: Option 3 also displays the error message as intended.

All is working as planned! From here, it's now time for us to enter our data. So, let's choose Option 1. (Image 2.4)

```
Please enter a menu option from the list above: 1
Please enter the following information to register a new student.

Please enter the student's first name: Jesse
Please enter the student's last name: Huminski
Please enter the course name you wish to register for: Python 100

Congratulations, Jesse Huminski has successfully registered for Python 100!

----Course Registration Program----

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above: |
```

Image 2.4: Entering the user data. A success message is printed and the program loops back to the beginning.

Now, let's check to see if the program displays the collected information back to the user by selecting option 2. (Image 2.5)

```
Please enter a menu option from the list above: 2

Jesse,Huminski,Python 100

----Course Registration Program----

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above:
```

Image 2.5: CSV Formatted data collected from the user displays properly.

Looking good there! Note how the collected information is displayed back to the user as opposed to the user receiving the error message from Image 2.2. Now, we need to save the collected data to the CSV file by choosing option 3. (Image 2.6)

```
Please enter a menu option from the list above: 3

Your request has been processed and saved.

----Course Registration Program----

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above:
```

Image 2.6: A success message is displayed informing the user their data is saved.

Looking good... let's see if the data was saved properly to the CSV file by opening it in Notepad. (Image 2.7)

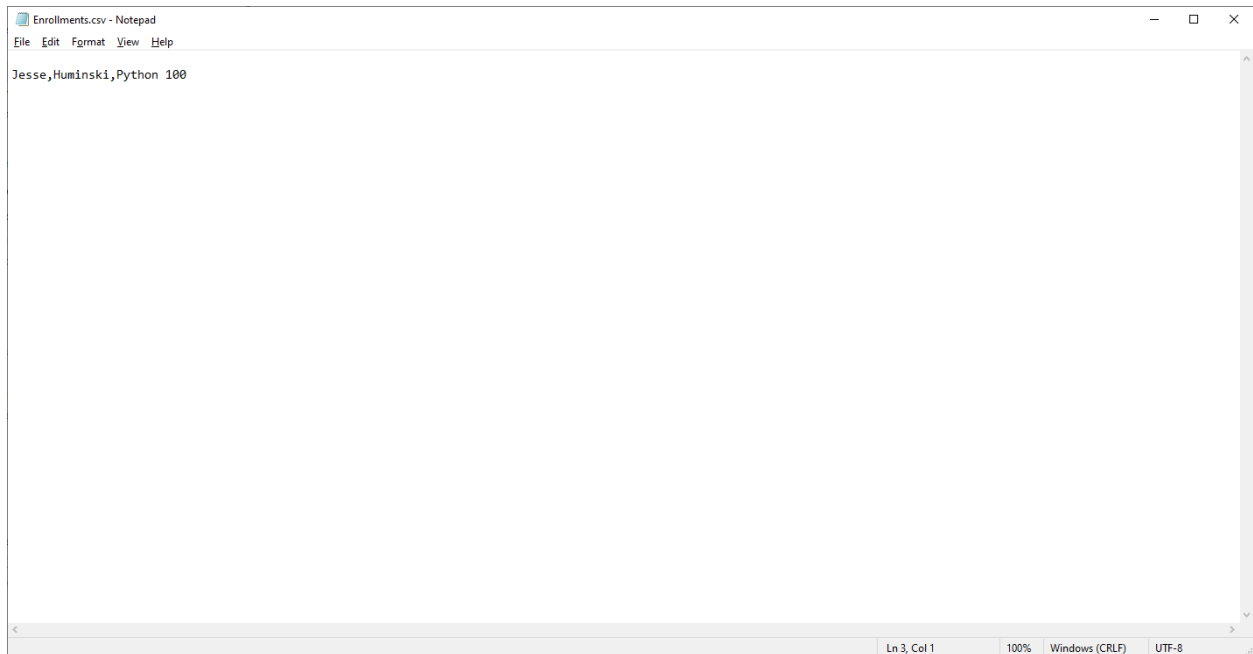


Image 2.7: CSV Data printed and saved successfully to the CSV file.

Before we exit, I just want to make sure that the code displays a message if an incorrect option is selected works as intended. Let's enter "5" when prompted for an option. (Image 2.8)

```
Please enter a menu option from the list above: 5

Invalid option. Please read the menu and choose an applicable option.

----Course Registration Program----

Select from the following menu:

1. Register a student for the course
2. Show current data
3. Sava data to a file
4. Exit the program
-----

Please enter a menu option from the list above:
```

Image 2.8: Invalid option message prints properly.

Awesome! All is looking to work as intended. Now we need to exit the loop. Let's select option "4" to see if the program responds appropriately. (Image 2.9)

```
Please enter a menu option from the list above: 4

You have exited the program

Process finished with exit code 0
```

Image 2.9: Program (and loop) have successfully ended.

Finally, let's run this in the command shell (Image 2.10)

```
Microsoft Windows [Version 10.0.19045.5811]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jesse>cd C:\Users\Jesse\Documents\Python Course\Module 3\A03
C:\Users\Jesse\Documents\Python Course\Module 3\A03>python Assignment03.py
----Course Registration Program----

Select from the following menu:
1. Register a student for the course
2. Show current data
3. Save data to a file
4. Exit the program
-----
Please enter a menu option from the list above: 1
Please enter the following information to register a new student.
Please enter the student's first name: Jesse
Please enter the student's last name: Huminski
Please enter the course name you wish to register for: Python 100
Congratulations, Jesse Huminski has successfully registered for Python 100!

----Course Registration Program----

Select from the following menu:
1. Register a student for the course
2. Show current data
3. Save data to a file
4. Exit the program
-----
Please enter a menu option from the list above: 2
Jesse,Huminski,Python 100

----Course Registration Program----

Select from the following menu:
1. Register a student for the course
2. Show current data
3. Save data to a file
4. Exit the program
-----
Please enter a menu option from the list above: 3
Your request has been processed and saved.

----Course Registration Program----

Select from the following menu:
1. Register a student for the course
2. Show current data
```

Image 2.10: Running the program in the Command Shell

Summary

Assignment 03 expands our rudimentary knowledge of Python by requesting us to create menus for the user to interact with and utilize loops to have the user go back to the option selecting screen. We utilize knowledge we learned from Modules 01 and 02 in this code and expand a bit further to allow the program to be more interactive. We continue to use variables and constants and incorporate "Match and Case" statements, the "While" loop and "Continue" and "Break" so that the user is able to reiterate the code should they need to and exit the code when they wish to finish.