Callie Weber
5/12/2024
Foundations of Python Programming
Assignment 05

Creating a Python Script

Introduction

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 Assignment03, but it adds the use of data processing using dictionaries and exception handling

Drafting the Code

The first thing I did was create my header text so I could track changes and define the purpose of the script. Next, I imported my necessary libraries and I defined my constants' and variable data types and values where applicable per the directions for this assignment. I could not find any reason to use "csv_data" within my script.

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment04
# Desc: This assignment demonstrates using constants, variables, and print()
# Change Log: (Who, When, What)
#   cweber,5/12/2024,Created Script
# ------------------------------------------------------------------------------------------ #

from sys import exit

# define 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
-------------------------------------"""
# print(MENU)
FILE_NAME = "Enrollment.csv"

# variables:
student_first_name: str = ''
student_last_name: str = ''
course_name: str = ''
csv_data: str = ''  # I did not find a reason to use this
file_obj = None
menu_choice: str = ''
student_data: dict
students: list = []
parts: list[str]
```

Next I read my current csv file and created my lists and dictionaries. I added an exception for if the file was not available and could not be found:

Callie Weber
5/12/2024
Foundations of Python Programming
Assignment 05

```
try:
    file = open(FILE_NAME, 'r')
    for row in file.readlines():
        parts = row.strip().split(',')
        student_first_name = parts[0]
        student_last_name = parts[1]
        course_name = parts[2]
        student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': course_name}
       # student_data = row.strip().split(',') --> leftover from assignment04
        students.append(student_data)
except FileNotFoundError as e:
    print('Text file not found')
    print('---Technical Information---')
    print(e, e.__doc__, type(e), sep='\n')
finally:
    if not file.closed:
        file.close()
print(students)
```

Next, I created my while loop for each menu option. Option 1 is defining the inputs of the student and class data and putting the data into the “student_data” list. Then the individual students in the list are appended into the full “students” dictionary. I also added exceptions that first and last names must be alphabetical.

```
if menu_choice == "1":
    try:
        student_first_name = input("Please enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError('The first name must be alphabetic')
        student_last_name = input("Please enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError('The last name must be alphabetic')
        course_name = input("Please enter the course name: ")
        student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': course_name}
        students.append(student_data)
    except ValueError as e:
        print(e)
        print('---Technical Information---')
        print(e, e.__doc__, type(e), sep='\n')
```

Option 2 prints all the students in the “student_data” as a string:

```
# print a string of the student data list
    elif menu_choice == "2":
        print("-" * 50)
        for student_data in students:
            print(f"Student {student_data['first_name']} {student_data['last_name']} is enrolled in {student_data['course_name']}")
        print("-" * 50)
        continue
```

Option 3 is where we write our dictionary values into our csv file. I added two exceptions for writing to the csv:

Callie Weber
5/12/2024
Foundations of Python Programming
Assignment 05

```
elif menu_choice == "3":
    try:
        file = open(FILE_NAME, 'w')
        for student_data in students:
            student_first_name = student_data['first_name']
            student_last_name = student_data['last_name']
            course_name = student_data['course_name']
            file.write(f'{student_first_name},{student_last_name},{course_name}\n')
        file.close()
    except TypeError as e:
        print('csv data was malformed')
        print('---Technical Information---')
        print(e, e.__doc__, type(e), sep='\n')
    except Exception as e:
        print('---Technical Information---')
        print(e, e.__doc__, type(e), sep='\n')
    finally:
        if not file.closed:
            file.close()
```

Lastly, Option 4 closes the program and I added an “else” statement to catch all mistaken entries.

Summary

I still don’t love the “student_data” and “student” names as they aren’t clear which is which, I would have chosen different names. This assignment was much easier as we continued to build on last week’s code.