Jennifer Hernandez-Mora

May 29, 2024

Foundations of Python Programming

Github Assignment Repository: https://github.com/jenhzm

Assignment 5 - Advanced Collections and Error Handling

Introduction

In this document, I will outline my steps to continue building a Python script in Pycharm that displays a student's class registration information. This assignment builds on assignment 4 and the end goal is to display a message about a student's registration information for a course. However, this assignment has the additional requirement of using a list of dictionaries as opposed to lists from assignment 4. There was also an added element of implementing

Organizing Code Format

Given that assignment 5 builds on 4, much of the program remained the same in terms of content. However, as I wanted to use dictionaries instead of lists, I identified what needed to be tweaked to implement this. In addition, for this assignment, the use of JSON was a requirement so that was a challenge as I had not worked with this type of file before.

Drafting the Code

The data variables and constants, in student_data, were updated to type dictionary versus list. Further, with the utilization of a JSON file as opposed to a CSV, there were needed changes to make when it came to opening the JSON file. Another implemented change that improved the program was the incorporation of error handling. This was done in the form of entering “try”, “except”, and “exception” as well as “raise”. An example of this can be seen in Figure 1. This also prompted the use of “isalpha” which ensured that entered values would meet proper criteria and if not then it would notify the user to fix this.

```
# Input user data
if menu_choice == "1":  # This will not work if it is an integer!
    try:
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError("Student First Name must be alphabetic")
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("Student Last Name must be alphabetic")
        course_name = input("Please enter the name of the course: ")
        student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': course_name}
        students.append(student_data)
        print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
    except ValueError as e:
        print(e)
    except Exception as e:
        print("Error with entered data")
        print(e.__doc__)
    continue
```

Summary

In the end, I was able to read an existing file, get data from the user, display the data, and save data to the file. However, there were added elements which made this program more robust. The use of error handling is useful so that if the user enters inputs that would not meet the requirements then the user is prompted to try again. Additionally, I learned about JSON files and implemented one in this code.