Assignment 5 – Advanced Collections and Error Handling

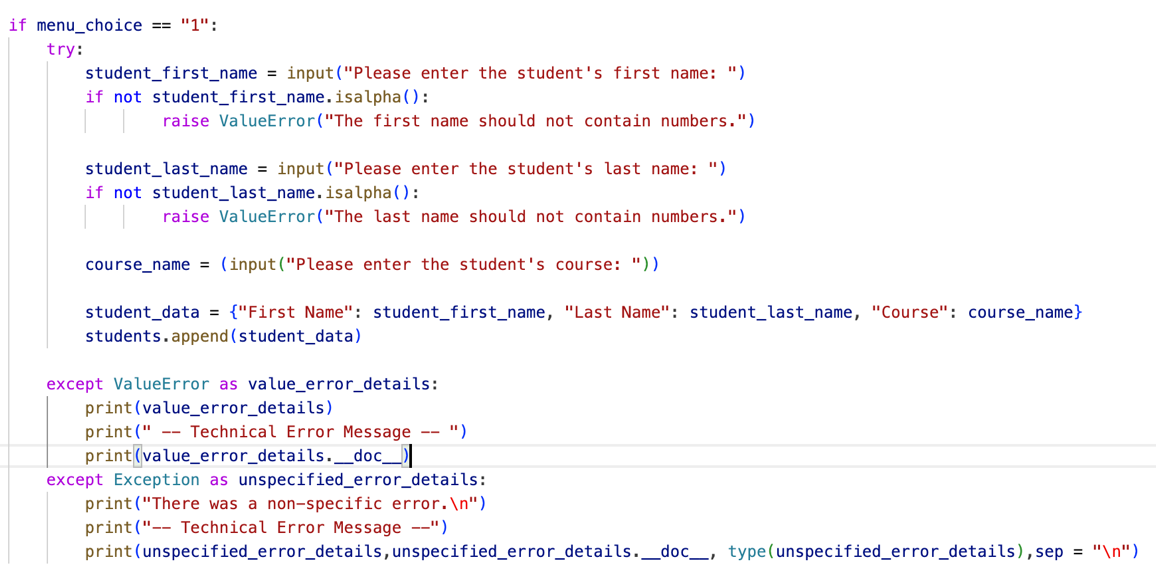
Introduction

In this assignment, I created a menu and wrote code to execute the functions of the menu. I used dictionaries and a table (list of lists) to organize my data. I also used try-catches to handle errors I anticipated users may encounter while running the code.

Creating the Program

I used the constants (FILE\_NAME and MENU) which we have been using throughout the course. I also used the global variables which the assignment suggested. I didn’t end up using the “csv\_data” string variable because I couldn’t figure out how to incorporate that into my code.

The first option asked users to enter the students’ first and last names, and the course they were registering for. I compiled this information into a dictionary entry “student\_data” and appended the dictionary to the table “students.” I used a try-except statement to catch errors in user’s inputs. If users input any numbers while they were giving student names, the ValueError would be thrown. I used a generic error message to catch unanticipated errors.



The second option allowed users to see all the registrations that had been made. I had a lot of trouble with this code and it still doesn’t run correctly. I wrote this code so that it would upload the data from the CSV file and would sort it according to dictionary keys and to create dictionary entries. Then, I appended those dictionary entries to the table “students”. Next, I printed the table “students” by pulling the values from the dictionary keys and converting them to strings, to print them out. I added the try-except statements to catch FileNotFound and non-specified errors. the FileNotFound error will be thrown if someone has not yet created the CSV file and tries to run #2. The non-specified error will catch unanticipated errors.

This code results in double-printing, however. I know that the “students” table is collecting more information than I want it to, but I don’t know how to prevent it from collecting repeat information. I have tried to solve this in several ways but so far nothing has worked.



The third option allows users to save the input data (from #1) to file. This option works well. I added a try-except statement to catch TypeErrors and non-specified errors. The TypeError will be thrown if someone tries to save data that is not in a proper CSV format. The non-specified error will catch unanticipated errors.

A computer code with text

Description automatically generated

The fourth option allowed users to leave the menu system. I used a *break* command.

A close-up of a word

Description automatically generated

Finally, I added an else statement in case users pressed any number that wasn’t on the menu. It displayed a message asking them to choose an option 1-4.

Testing the Program

I tested this code on the Python IDE and the Python terminal. In both cases, it partially works. The #1,3,4,5 options work but option #2 does not work as I would like (more description given above). I tried to solve this problem using several resources but haven’t been able to get it to run as I would like.

Summary

I created a *while loop* to guide users through four menu options, used dictionaries and lists to organize the data, and embedded try-except statements to catch any errors users may encounter while using the script. I didn’t get the script working 100% correctly but it does perform the essential functions.