Jorden Thomas

Feb 19, 2025

IT FDN 110

Assignment 05

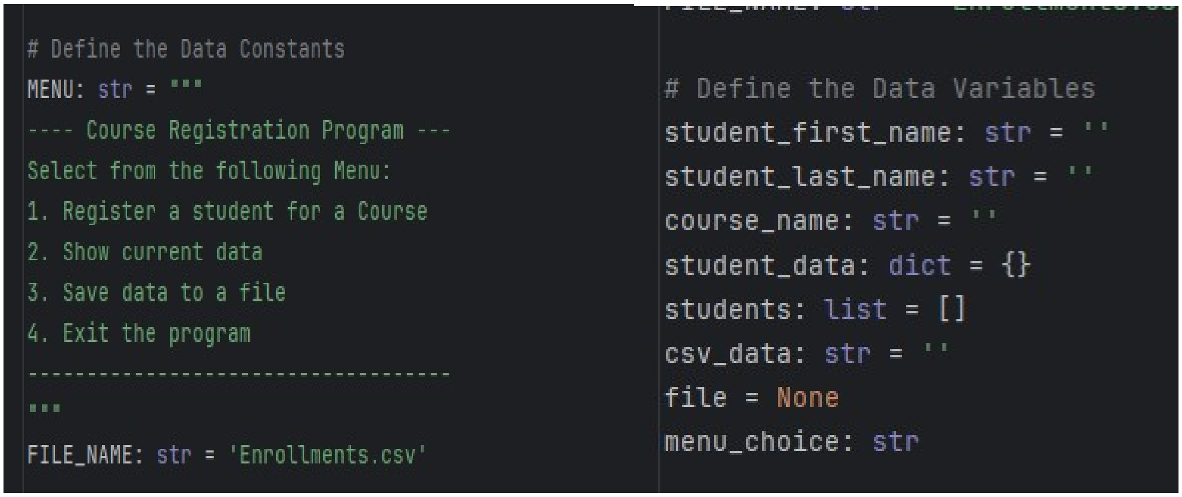
https://github.com/Jorden09T/IntroToProg-Python-Mod05

Assignment 05

Introduction

For this assignment, the focus was on expanding Assignment04 by adding to the use of data processing, dictionaries, and exception handling.

Constants & Variables

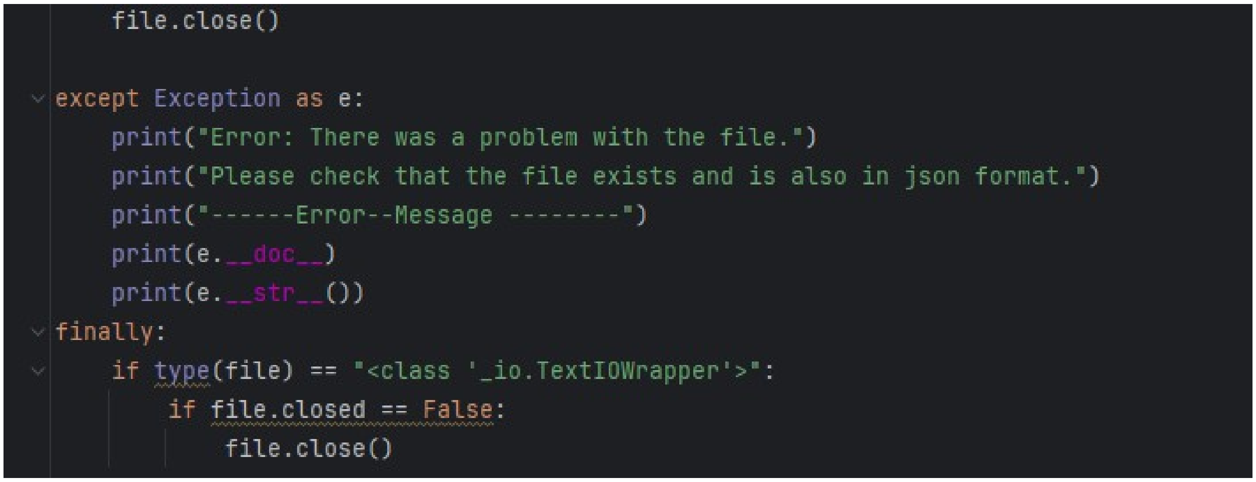


```
# Define the Data Constants
MENU: str = """
---- Course Registration Program ---
Select from the following Menu:
1. Register a student for a Course
2. Show current data
3. Save data to a file
4. Exit the program
-----------------------------------
"""
FILE_NAME: str = 'Enrollments.csv'
```

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

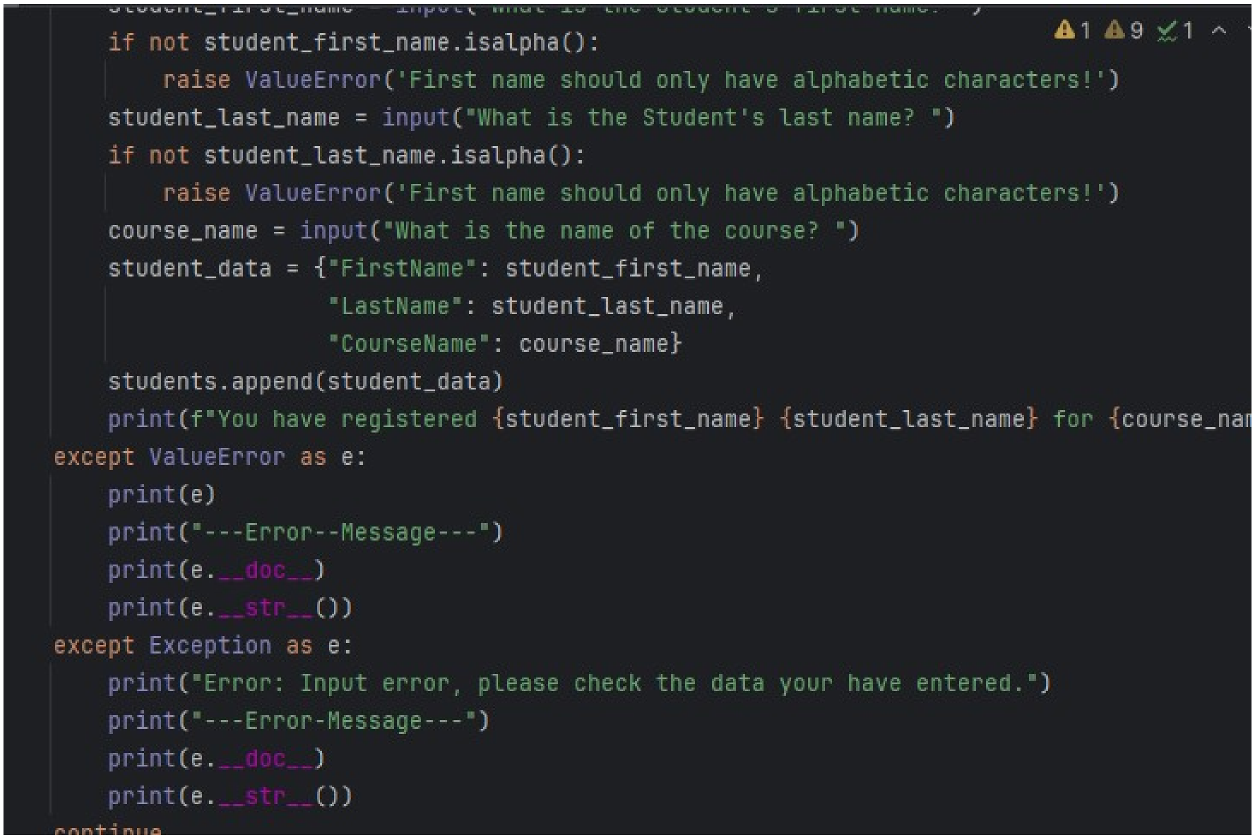
Nothing changed for the constants, they remained as they were. As for, the variable section, the only thing to change was the student _data becoming a dict instead of a list and file_obj to be file. Other than those changes nothing changed for the constants.

Exceptions



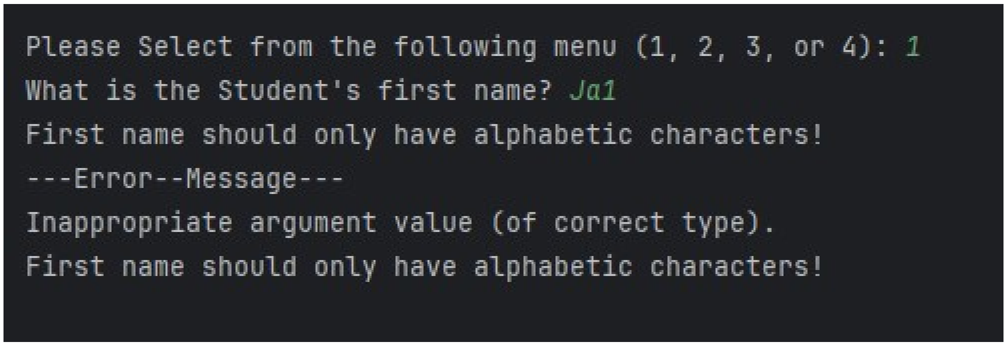
```
    file.close()

except Exception as e:
    print("Error: There was a problem with the file.")
    print("Please check that the file exists and is also in json format.")
    print("------Error--Message --------")
    print(e.__doc__)
    print(e.__str__())
finally:
    if type(file) == "<class '_io.TextIOWrapper'>":
        if file.closed == False:
            file.close()
```

For the exceptions there were a total of 3 added through out the code the first was when checking for the required file which was the enrollment.csv. it checked to see if the file was present and if it were closed. If the file was not present then an error message was displayed to show that the file was missing. Which is good because on the last assignment I accidentally deleted mine some how.



```
    student_first_name = input("What is the Student's first name? ")
    if not student_first_name.isalpha():
        raise ValueError('First name should only have alphabetic characters!')
    student_last_name = input("What is the Student's last name? ")
    if not student_last_name.isalpha():
        raise ValueError('First name should only have alphabetic characters!')
    course_name = input("What is the name of the course? ")
    student_data = {"FirstName": student_first_name,
                    "LastName": student_last_name,
                    "CourseName": course_name}
    students.append(student_data)
    print(f"You have registered {student_first_name} {student_last_name} for {course_nar
except ValueError as e:
    print(e)
    print("---Error--Message---")
    print(e.__doc__)
    print(e.__str__())
except Exception as e:
    print("Error: Input error, please check the data your have entered.")
    print("---Error-Message---")
    print(e.__doc__)
    print(e.__str__())
continue
```

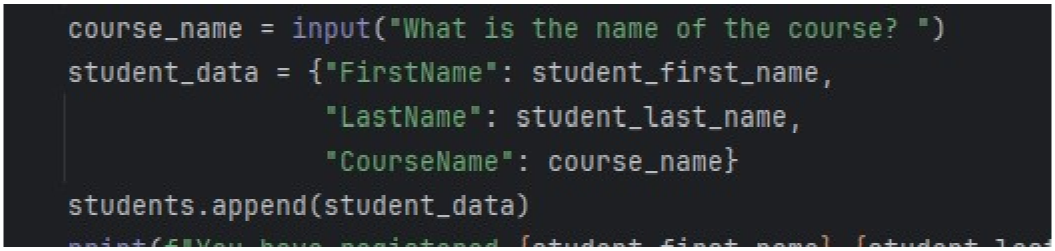
The next exception checked to see if the correct characters were used when entering in the student's first and last name. More specifically, whether there were numbers present. Which works as shown below. When I entered in the first part of the name and added in a 1 before pressing enter to send off my input to the program.



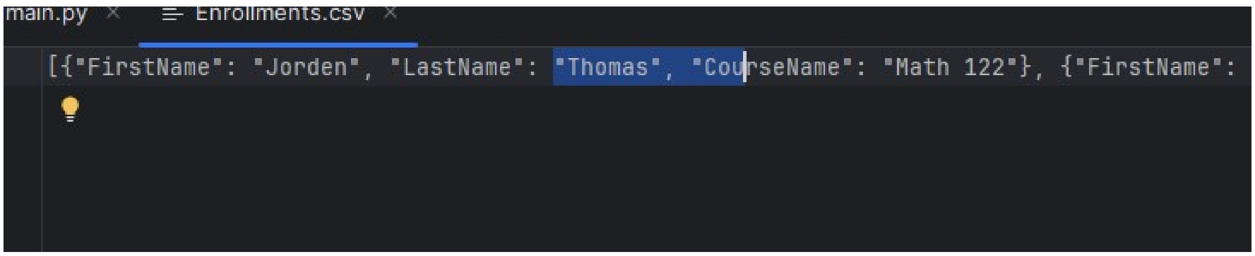
```
Please Select from the following menu (1, 2, 3, or 4): 1
What is the Student's first name? Ja1
First name should only have alphabetic characters!
---Error--Message---
Inappropriate argument value (of correct type).
First name should only have alphabetic characters!
```

The last exception is for option 3) save data to a file. When a file is opened in another program an error message is supposed to showcase this error so the user can know to close the program in order to save the file.

Dictionaries



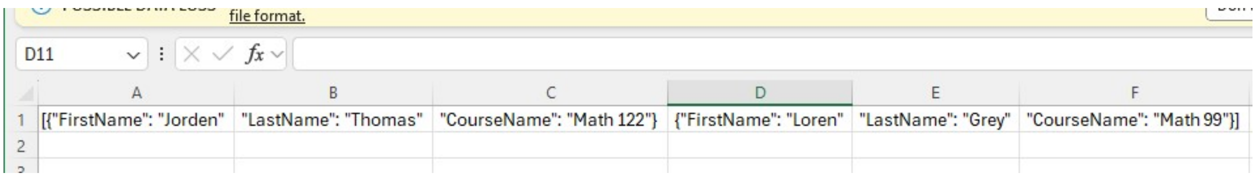
```
    course_name = input("What is the name of the course? ")
    student_data = {"FirstName": student_first_name,
                    "LastName": student_last_name,
                    "CourseName": course_name}
    students.append(student_data)
    print(f"You have registered {student_first_name} {student_last
```



```
[{"FirstName": "Jorden", "LastName": "Thomas", "CourseName": "Math 122"}, {"FirstName":
```

For the dictionary portion of the code, we are storing key-value pairs, to have a more organized format. For this assignment, in the first photo, we can see the data is being stored with “FirstName” being the student's first name, “LastName” the student's last name, and “CourseName” the name of course the student has registered for. In the second photo, we can see a snippet of the stored data with the first name, last name, and course name appearing before the next value pair of information for the next student.

Output Results



	A	B	C	D	E	F
1	[{"FirstName": "Jorden"	"LastName": "Thomas"	"CourseName": "Math 122"}	{"FirstName": "Loren"	"LastName": "Grey"	"CourseName": "Math 99"}]
2						

The only change about the output results that have changed is how the enrollment.CSV file now looks when opening it in Excel. As we can see the data is kept in the same format that we listed for the dictionary that we assigned for student_data. They are no longer by row but in the same row.