Name: Blaise Konzel

Date: May 28, 2025

Course: Foundation of Programming (Python): Python 100

Assignment 06

Introduction

This assignment continued working with JSON files as well as including provisions for error handling during the use of the course registration database code and added in functions and classes to better structure and segment the working program.

Code

1) Defining constants and variables

```
# Define the Constants and Variables ------------------
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.json"
students: list = []  # a table of student data
menu_choice: str = ''  # Hold the choice made by the user.
```

As opposed to previous assignments, now we are initializing the bare minimum amount of variables that will be used in this program. Because we have moved most of our working code into functions, we no longer need to store as many single variables but instead keep everything in a few larger variables. For example, instead of storing first, last and course names, we now are assigning them right after they are defined and then erase their stored value as soon as the function breaks. The only variables defined are ‘MENU’, ‘FILE_NAME’, ‘students’, and ‘menu_choice’.

2) Defining class **IO**

In this class I have defining using static methods the functions:

- output_error_messages- This function handles all error messaging output include custom messages as well as error type information.
- output_menu- This function simply outputs the standard MENU
- input_menu_choice- Function takes input choice from the user and calls associated function
- output_student_courses- prints dictionary list to screen
- input_student_data-allows the user to register a new student and adds info to the current list

3) Defining class **FileProcessor**

In this class I have defining using static methods the functions:

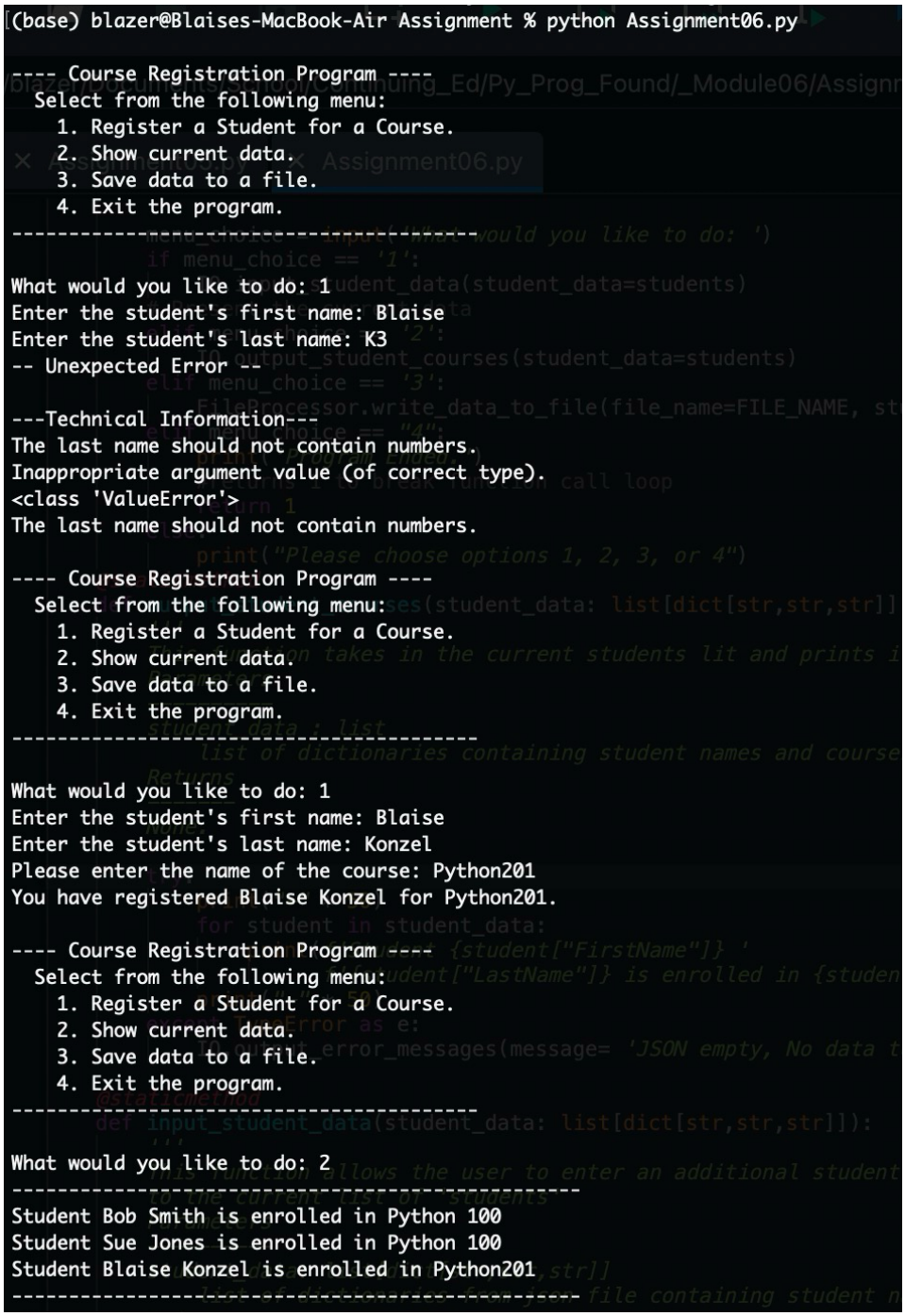
- read_data_from_file- reads data from the JSON file. Will create file if one is not present (on purpose)
- write_data_to_file- writes the current list of dictionaries stored in 'students' to a JSON file named Enrollments.JSON.

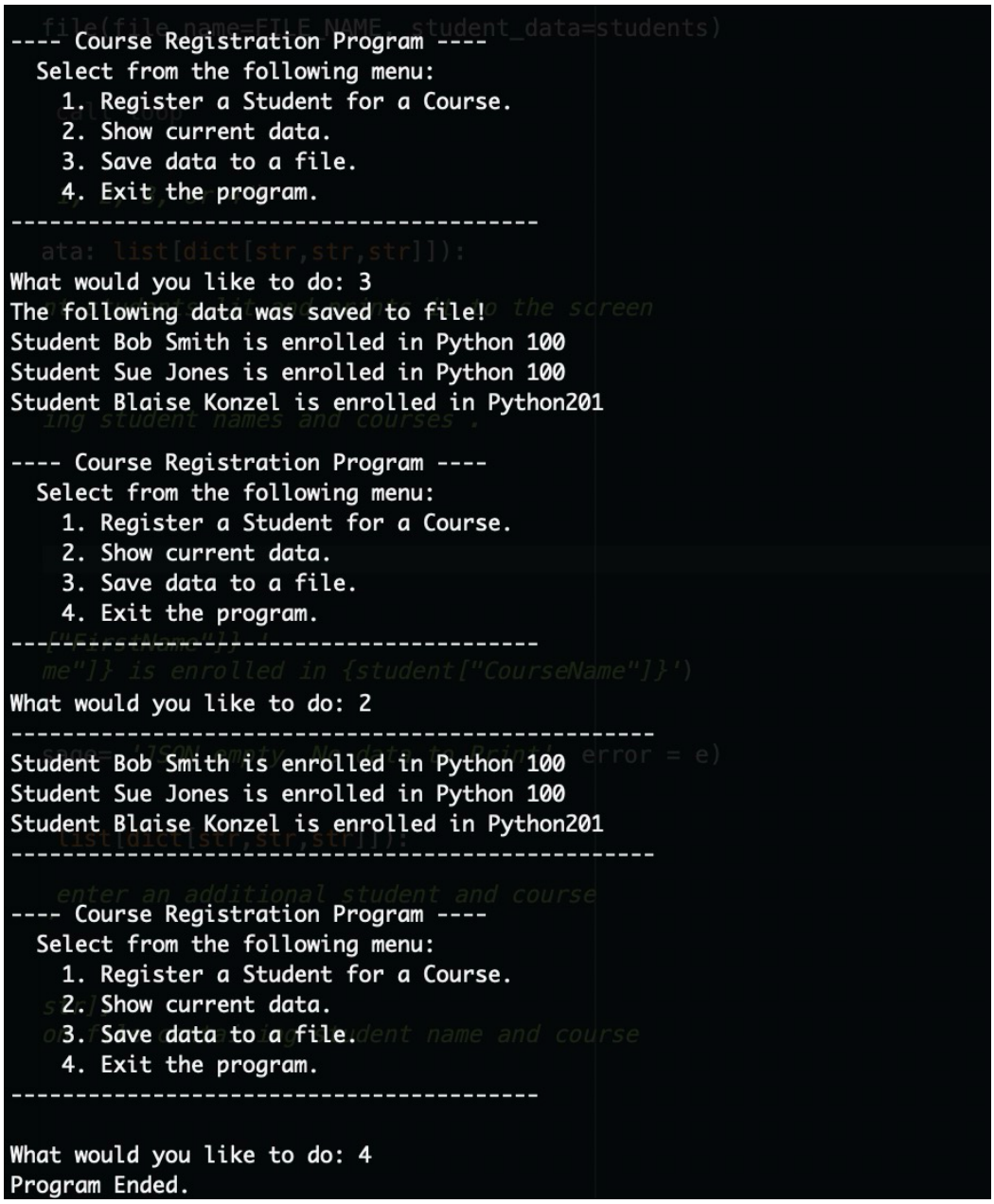
4) Program Run

```
#%% running the program

# populate the 'students' var
students = FileProcessor.read_data_from_file(student_data=students, file_name=FILE_NAME)

#intialize var to break program loop
loop_breaker = 2
#program containing loop
while loop_breaker !=1 :
    IO.output_menu(MENU)
    loop_breaker = IO.input_menu_choice()
```

The above code snippet contains all that is needed to run the program as its defined in the classes/functions I described earlier. You'll notice that I employed a 'loop_breaker' variable. Simply, any time the function IO.input_menu_choice() returns a 1 (when user selects 4) the code breaks the while loop. I didn't have the same success using True/False Booleans.

Running the Code from Terminal





This terminal snippet shows the code taking new data from the user, displaying current data, and successfully writing the data to my local copy of Enrollments.json.

Summary

This assignment was a great way to learn class and function building. It took some focus to understand how the code should be organized and structured in a way that was easy to read and use. At times it was difficult tracking which functions called which other functions. The error handling function was the most difficult for me to understand. I struggled to incorporate the correct technique when using the ‘raise’ command.