Name: Philip Liu
Date: February 21, 2024
Course: IT FDN 110 A

Assignment06 – Course Registration Program Menu with Functions, Classes, SoC Methodology

Intro

In our continuance of providing the best customer experiences of our Course Registration Program menu product line, we now have iterated our product with new Functions, Classes, and implementation of the SoC methodology to further solve front-end and back-end pain-points. Newly added features will be marked, for example: *Functions*.

1.1: SoC, Data Layer. Accessing Data at Rest and in Different Forms. Read & Write to File is Wrapped for Error Handling

```
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"  # File name Constant

student_data: list = []  # List of data entered by user
menu_choice: str = ''  # Hold the choice made by the user
```

Continue to Page 2.

```python
@staticmethod
def read_data_from_file(file_name: str):
    """
    Function 1: Read data from json file
    """
    try:
        with open(file_name, "r") as file:
            student_data = json.load(file)
            return student_data
    except FileNotFoundError:
        IO.output_error_messages("File must be present")
        return []
    except json.JSONDecodeError:
        IO.output_error_messages("File must contain valid JSON")
        return []
    except Exception:
        IO.output_error_messages("Error!")
        return []

@staticmethod
def write_data_to_file(file_name: str, student_data: list):
    """
    Function 2: Write data to json file
    """
    try:
        with open(file_name, "w") as file:
            json.dump(student_data, file)
    except FileNotFoundError:
        IO.output_error_messages("Ensure correct file type")
```

Continue to Page 3.

1.2: SoC, Presentation Layer. Classes IO, along with Their Respective *Functions*

```
### Prestation Layer
class IO:
    """
    ChangeLog: Philip Liu. 2/21/2024. Created Class
    Collection of input, output, display functions to communicate with user
    """

    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        print("Error:", message)

    @staticmethod
    def output_menu():
        """
        Display menu of choices
        """
        print(MENU)

    @staticmethod
    def input_menu_choice():
        """
        Prompts user for input
        """
        return input("What would you like to do? ")

    @staticmethod
    def input_student_data():
        """
        Get student data from the user
        """
        try:
            student_first_name = input("Enter first name: ")
            if not student_first_name.isalpha():
                raise ValueError("No numbers, please. ")
            student_last_name = input("Enter last name: ")
            if not student_last_name.isalpha():
                raise ValueError("No numbers, please. ")
            course_name = input("Enter course name: ")
            student_data = {"First Name": student_first_name, "Last Name": student_last_name,
                            "Course": course_name}
            return student_data
        except ValueError as e:
            print("Error:", e)
            return None
```

Continue to Page 3.

```python
    @staticmethod
    def output_student_data(student_data: list):
        """
        Displays user's data
        """
        for student in student_data:
            if all(key in student for key in ["First Name", "Last Name", "Course"]):
                print(f'{student["First Name"]} {student["Last Name"]} is enrolled into {student["Course"]}')
            else:
                print("Invalid student data:", student)
        print("-" * 50)
```

1.3: SoC, Application Layer. *Functions* with Their Associated Logic Wrapped In Loop for Continuance of Service Until Exit.

```python
### Application Layer
def main():
    student_data = FileProcessor.read_data_from_file(FILE_NAME)

    while True:

        IO.output_menu()
        menu_choice = IO.input_menu_choice()

        if menu_choice == "1":
            student = IO.input_student_data()
            if student:
                student_data.append(student)

        elif menu_choice == "2":
            print("\nThe current data is: ")
            IO.output_student_data(student_data)
            IO.output_menu()

        elif menu_choice == "3":
            FileProcessor.write_data_to_file(FILE_NAME, student_data)
            print("Data saved to file")

        elif menu_choice == "4":
            print("Exiting program...")

            break
        else:
            print("Please only choose options 1, 2, 3 or 4")


if __name__ == "__main__":
    main()
```

Conclusion

Our latest iteration of previous code of Course Registration Program Menu, propels our code of service to meet industry standards, SoC. Layering the process of how our code provides services to better readability, maintainability, and scalability; while reducing downtime and potential bugs.
Also, implementing Class and their respective Functions enables us to modularize and reuse our code, decreasing redundancy.