Name: Mike Luke

Date: 11/17/2025

Course: IT FDN 110 A Au 25: Foundations Of Python Programming

Assignment 06 –

Introduction

For Week 6, we are introduced to classes, functions, parameters, arguments, global variables and separation of concern. It is a lot of great content and concepts this week. It is starting to make a bit more sense and introduces organization with classes, functions, and separation of concern.

Topic

Starting off I began, by writing in my pseudo code to help me organize the flow. Once I had that laid in, I wrote out each of the function names.

```
    # Read from File
    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):  1 usage (1 dynamic)

    @staticmethod

# Error exception message
    @staticmethod
    def output_error_messages(message: str, error: Exception = None):

    @staticmethod  1 usage (1 dynamic)
    def input_menu_choice(menu_choice: str):

    # Show Menu
    @staticmethod  1 usage (1 dynamic)
    def output_menu(menu: str):

    # User Input
    @staticmethod  1 usage (1 dynamic)
    def input_student_data(student_data: list):

    # Show list contents
    @staticmethod  1 usage (1 dynamic)
    def output_student_courses(student_data:list):
```

Then I organized with classes by what their functions were or handled the data.

```
# Presentation
class FileProcessor:  2 usages
    # Read from File
    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):  1 usage

    # Save data to file
    @staticmethod  1 usage
    def write_data_to_file(file_name: str, student_data: list):

class IO:  4 usages
    pass

# Error exception message
    @staticmethod
    def output_error_messages(message: str, error: Exception = None):

    # Input Menu for user
    @staticmethod  1 usage
    def input_menu_choice(menu_choice: str):

    # Show Menu
    @staticmethod  1 usage
    def output_menu(menu: str):

    # User Input
    @staticmethod  1 usage
    def input_student_data(student_data: list):

    # Show list contents
    @staticmethod  1 usage
    def output_student_courses(student_data:list):
```

Having all my classes and functions organized made it easy to work through the code. We have been using similar if not very much the same code for a few weeks now. The difference this time is that we used functions inside of classes to do the repetitive work.

```python
class IO:  11 usages
    pass

# Error exception message
    @staticmethod  7 usages
    def output_error_messages(message: str, error: Exception = None):

        print(message, end="\n\n")
        if error is not None:
            print("-- Technical Error Message -- ")
            print(error, error.__doc__, type(error), sep='\n')

    # Input Menu for user
    @staticmethod  1 usage
    def input_menu_choice(menu_choice: str):
        try:
            menu_choice = input("What would you like to do: ")
            if menu_choice not in ("1", "2", "3", "4"):
                raise Exception ("Please, Choose only 1, 2, or 3")
        except Exception as e:
            IO.output_error_messages(e.__str__())
        return menu_choice

    # Show Menu
    @staticmethod  1 usage
    def output_menu(menu: str):
        print()
```

After formatting all of the functions, I started to work on the program. This was great, because it was so concise and clear once it was completed with the classes/function/arguments.

```python
# Present and Process the data

while (True):
    # Present the menu of choices
    IO.output_menu(menu=MENU)

    menu_choice = IO.input_menu_choice(menu_choice=menu_choice)

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        IO.input_student_data(student_data=students)
        continue

    # Present the current data
    elif menu_choice == "2":

        # Process the data to create and display a custom message
        IO.output_student_courses(student_data=students)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
        continue

    # Stop the loop
    elif menu_choice == "4":
```

Summary

All in all this week's module was great. It really starts to help us organize and keep everything in focus. It also gets us start thinking more deeply on how to organize, the steps required, and thoughtful names for classes and functions. This will allow us to produce a well written and understandable code base, while eliminating time by re-using functions that are pre-determined early on in our programs.