Andrew King

November 22, 2023

IT FDN 110A

Assignment 06

Working with Functions

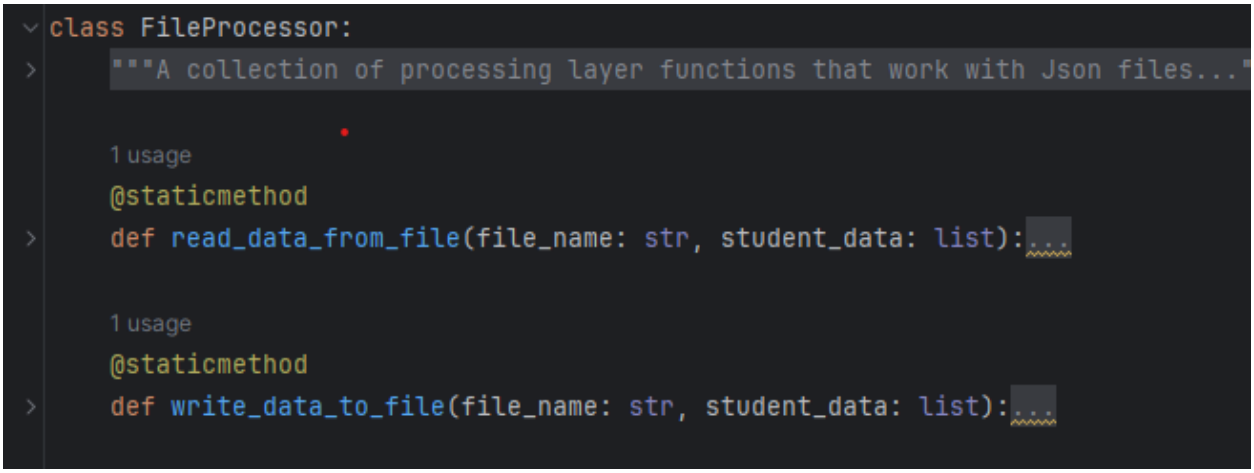
Introduction

This document explains the steps taken to complete assignment 6. The goal of this assignment was to incorporate functions, classes, and parameters into the code that was used to complete assignment 05, to complete the same task.

Completing the assignment

Creating classes and functions

In this assignment, functions are used to carry out various tasks within the code. These functions can be organized by placing them into classes, which store each function. For this assignment there were two classes, the FileProcessor calss and IO class. Filesprocessor worked with the functions that focused on reading and writing data to the json file. The IO class contained the functions that involved receiving input from the user and outputting data to the user. The two classes and the functions that belong to them are shown in Figure 1.



```
class IO:
    """A collection of presentation layer functions that manage user input and output..."""

    1 usage
    @staticmethod
    def output_menu(menu: str):...

    1 usage
    @staticmethod
    def input_menu_choice():...

    1 usage
    @staticmethod
    def input_student_data(student_data: list):...

    1 usage
    @staticmethod
    def output_student_courses(student_data: list):...

    @staticmethod
    def output_error_messages(message: str, error: Exception = None):...
```

Figure 1: Script Classes

The code beneath each of these functions performs the action it is named after. These perform the same actions that Assignment 5 did, however this format is much more concise and neater. The code within the script are the parameters used in each function. When the function is called later on, these parameters are given values, as shown in the red font. The function calls are shown in figure 2.

```
# Extract the data from the file

students = FileProcessor.read_data_from_file(file_name=FILE_NAME, student_data = students)

# -- Presentation (Input/Output) -- #
    # Present and Process the data
while (True):

    # Present the menu of choices
    IO.output_menu(menu=MENU)
    IO.input_menu_choice()

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

    # Present the current data
    elif menu_choice == "2":
        IO.output_student_courses(student_data=students)
        continue

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

```
    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, or 3")

print("Program Ended")
```

Figure 2: Calling the functions

Here, the functions are called, which causes them to perform their specific action. The functions are ran in the specific order so that the program can run as intended. The results of running the program are shown in Figure 3.

```
---- 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.
----------------------------------------

Enter your menu choice number: 1

Enter the student's first name: Drew
Enter the student's last name: Johnson
Please enter the name of the course: Python 150
You have registered Drew Johnson for Python 150.
```

```
Enter your menu choice number: 2

-------------------------------------------------
Student Vic Vu is enrolled in Python 110
Student Bob Baker is enrolled in Python 120
Student Drew King is enrolled in Python 110
Student Drew Johnson is enrolled in Python 150
-------------------------------------------------
```

```
Enter your menu choice number: 3

The following data was saved to file!
Student Vic Vu is enrolled in Python 110
Student Bob Baker is enrolled in Python 120
Student Drew King is enrolled in Python 110
Student Drew Johnson is enrolled in Python 150
```

```
Enter your menu choice number: 4

Program Ended

Process finished with exit code 0
```

Figure 3: Output of Program

Conclusion

This was a very interesting module that we learned, as functions are a core part of programming. Learning about classes and parameters were also interesting, though they took a little longer for me to understand compared to functions. We also learned about Separation of concerns pattern, which will be a strategy for future coding that we do.