

Noah Jon

November 16, 2024

IT FDN 100 A

Assignment 06

Assignment 06: Functions

Introduction

Assignment 06's focus was on functions and integrating them into our code. The goal was to utilize these in tandem with the structured error handling that we learned about and used in the previous assignment. Another feature was the usage of a JSON file to store data instead of a CSV file.

Features in the Code

Constants and Variables

In this assignment, the same constants were used as Assignment 5 except the file extension is json instead of csv:

```
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.
-----
'''

# Define the Data Constants
FILE_NAME: str = "Enrollments.json"
```

This week's assignment had fewer variables than previously.

```
# Define the Data Variables
students: list = [] # a table of student data
menu_choice: str # Hold the choice made by the user.
file = None
```

Functions and Classes

This week's focus was on functions and classes. Classes are a collection of functions that are grouped together by their goal or focus. The two classes I used was `FileProcessor` and `IO`. `FileProcessor` included all of the functions that dealt with reading and writing data from

Noah Jon

November 16, 2024

IT FDN 100 A

Assignment 06

and to the JSON file. `IO` included all of the functions that worked with the inputs, outputs, and errors associated with them both.

```
@staticmethod
def read_data_from_file(file_name: str, student_data: list):
    try:
        file = open(file_name, "r")
        student_data = json.load(file)
        file.close()
    except FileNotFoundError as e:
        IO.output_error_messages("Text file must exist before running this script!", e)
    except Exception as e:
        IO.output_error_messages("There was a non-specific error!", e)
    finally:
        if file.closed == False:
            file.close()
    return student_data
```

This function read the data that was in the existing JSON file and wrote the data to a list, `student_data`.

```
@staticmethod
def write_data_to_file(file_name: str, student_data: list):
    try:
        file = open(file_name, "w")
        json.dump(student_data, file)
        file.close()
        IO.output_student_courses(student_data = student_data)
    except Exception as e:
        IO.output_error_messages("There was a problem with writing the data to the file", e)
    finally:
        if file.closed == False:
            file.close()
```

This function took the data that was on `student_data` and wrote it onto the JSON file.

```
@staticmethod
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")
```

This function was used whenever an error occurred. It would print out the equivalent error message and error data.

Noah Jon

November 16, 2024

IT FDN 100 A

Assignment 06

```
@staticmethod
def output_menu(menu: str):
    print()
    print(menu)
    print()
```

This function displays the menu choices to the user.

```
@staticmethod
def input_menu_choice():
    choice = "0"
    try:
        choice = input("Enter your menu choice number: ")
        if choice not in ("1", "2", "3", "4"):
            raise Exception("Please, choose only 1, 2, 3, or 4")
    except Exception as e:
        IO.output_error_messages(e.__str__())
    return choice
```

This function asks the user to input a menu choice number and registers that input.

```
@staticmethod
def output_student_courses(student_data: list):
    print("-"*50)
    for student in student_data:
        print(f"Student {student['FirstName']} {student['LastName']} is enrolled in " \
              f"{student['CourseName']}")
    print("-"*50)
```

This function displays the current student data that is saved in `student_data`, including the first, last, and course name of each student.

```
@staticmethod
def input_student_data(student_data: list):
    try:
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError("The first name should not contain numbers")
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("The last name should not contain numbers")
        course_name = input("Please enter the name of the course: ")
        student = {"FirstName": student_first_name,
                  "LastName": student_last_name,
                  "CourseName": course_name}
        student_data.append(student)
        print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
    except ValueError as e:
        IO.output_error_messages("One of the input values was the wrong type of data", e)
    except Exception as e:
        IO.output_error_messages("There was a non-specific error", e)
    return student_data
```

Finally, this function asks the user to input the first, last, and course name of the student and saves the inputs to `student_data`.

Noah Jon

November 16, 2024

IT FDN 100 A

Assignment 06

For every function that requires a user input or working with the JSON file, there are error messages which call on the error function to print the appropriate error statement and data.

Program Loop

Finally, the program loop calls each function when necessary. For example,

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

while True:
    # Presents menu choices
    IO.output_menu(menu=MENU)

    # Menu Choice Option is requested and processed
    menu_choice = IO.input_menu_choice()

    # Input user data
    if menu_choice == "1":
        students = IO.input_student_data(student_data=students)
        continue

    # Print current data
    elif menu_choice == "2":
        IO.output_student_courses
        continue

    # Write and save current data to JSON file
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(FILE_NAME, student_data=students)
        continue

    # End program
    elif menu_choice == "4":
        break

    else:
        print("Choose options 1, 2, 3, or 4")
```

The first step, similar to the previous assignments, is to read the existing data and pull it into a list. For menu choice 1, the program calls for the `input_student_data` function to start the process of inputting the student's data. For menu choice 2, the program calls for the `output_student_courses` function to display the current data. For menu choice 3, the

Noah Jon

November 16, 2024

IT FDN 100 A

Assignment 06

program calls for the `write_data_to_file` function to save the current data to the JSON file.

Menu choice 4 ends the program so that it does not loop indefinitely.

Conclusion

This assignment changed the structure of our previous assignments' code while keeping the goal the same. The utilization of functions helps organize all of the required programs into separate classes, and helps simplify calling each function when needed. Instead of repeating or copy and pasting the loops or commands when needed, one can simply call the function in its respective class.