Name: Kamaljeet Gill

Date :05/22/2024

Course: Foundation of Programming Python

Assignment 6: Functions

GitHub Link: https://github.com/gillkamal20/IntroToProg-Python-Module06

Topic covered in Module 6:

- This module started with concept Function. Function is a block of code that performs a specific task.
- In Python, an argument is the value passed to a function when it`s called. Fundamentally, parameters are the variables inside a function`s parentheses. Arguments provide values for those parameters.
- In Python, the return statement exits a function and returns the specified value to the caller. Multiple return statements may exist in a function, but only the one that fulfils the specified condition first is executed.
- Separation of Concern with Decorators We often see multiple functions with the same logic implemented explicitly.

Assignment

Started the assignment by using starter file and created class FileProcessor and IOProcessor are created.

In class FileProcessor, below functions are defined:

1) Read_data_from file () to load json file

```
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 Exception as e:
        print(e)
        print("File not found")
        print(e.__doc__)
        print(e.__str__())
    finally:
        if file.closed == False :
            file.close()
    return student_data
```

2) Write_data_to_file () : To dump the data in json file.

```
@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()
        print("Data written to file")
        for student in students:
            print(f'Student {student["FirstName"]} '
              f'{student["LastName"]} is enrolled in {student["CourseName"]}')
            print("-"*50)
    except Exception as e:
```

In class IOProcessor, below functions are defined:

1) Output_error_message() :

```
def output_error_message(message: str, exception: Exception = None):

    if exception is not None:
        print("-- Technical Error Message -- ")
        print(exception)  # Prints the custom message
        print(exception.__doc__)
        print(exception.__str__())

@staticmethod
def output_message(message: str):
    print(message)
```

2) Input_menu_choice()

```
@staticmethod
def input_menu_choice():
    menu_choice = input("Enter your menu choice number: ")
    if menu_choice not in ['1', '2', '3', '4']:
        IOProcessor.output_message("Please enter an option between 1 and 4")
    return menu_choice
```

Output

After successfully running the program, below are the outputs

Output 1: Register a new student

```
Enter your menu choice number: 1
Enter the student's first name: Ric
Enter the student's last name: Roy
Please enter the name of the course: Python
--------------------------------------------------
You have registered Ric Roy for Python.
--------------------------------------------------
```

Output 2: Show current Data

```
Enter your menu choice number: 2
--------------------------------------------------
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student kamal Jeet is enrolled in Excel
--------------------------------------------------
--------------------------------------------------
```

Output 3: Saving to the file

```
Enter your menu choice number: 3
Data written to file
Student Bob Smith is enrolled in Python 100
--------------------------------------------------
Student Sue Jones is enrolled in Python 100
--------------------------------------------------
Student kamal Jeet is enrolled in Excel
--------------------------------------------------
Student Ric Roy is enrolled in Python
--------------------------------------------------
```

Records successfully written in Enrollment.json file


```
[{"FirstName": "Bob", "LastName": "Smith", "CourseName": "Python 100"}, {"FirstName": "Sue", "LastName": "Jones", "CourseName": "Python 100"}, {"FirstName": "kamal",
"LastName": "Jeet", "CourseName": "Excel"}, {"FirstName": "Ric", "LastName": "Roy", "CourseName": "Python"}]
```

Summary

In this module, I have successfully used functions and arguments. I also learned how to use class and Separation of Concern. I still feel I need more practice to full grasp the concepts.