Name: Vugee Preap

Date: May 22, 2024

Course: IT FDN 110 A

Github

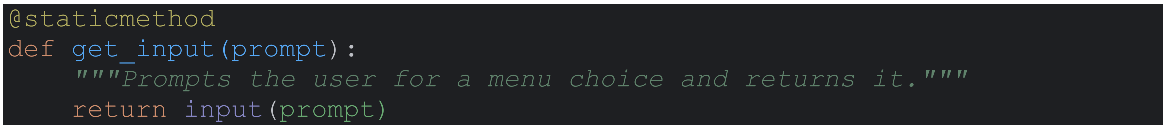
Module 06 – Functions

Introduction

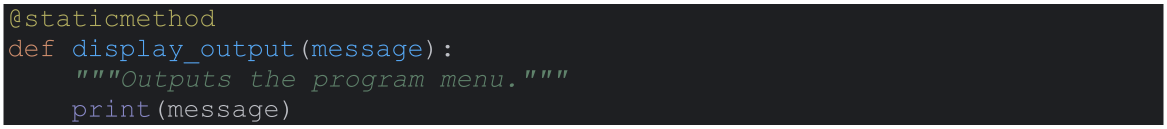
Module 06 covered Functions using Python script. This included learning of functions, parameters, and arguments as well as their similarities and usages. Lab 1 was on Functions, Lab 2 was on Parameters, and Lab 3 was Working with Classes and SoC. We finished the module by posting out code onto Github.

Define Classes and Methods

The first step to completing the assignment was to start by defining the two classes. The assignment required a class for FileProcessor and one for IO with all classes including descriptive document strings. Each class contains static methods for specific various task.



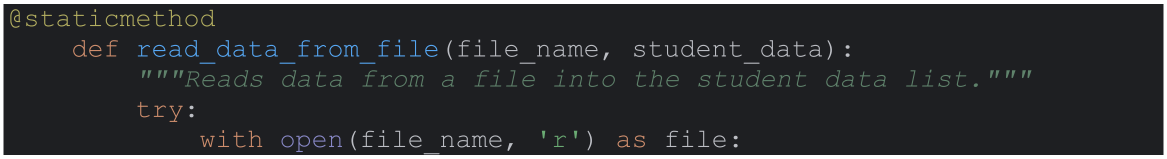
```
@staticmethod
def get_input(prompt):
    """Prompts the user for a menu choice and returns it."""
    return input(prompt)
```



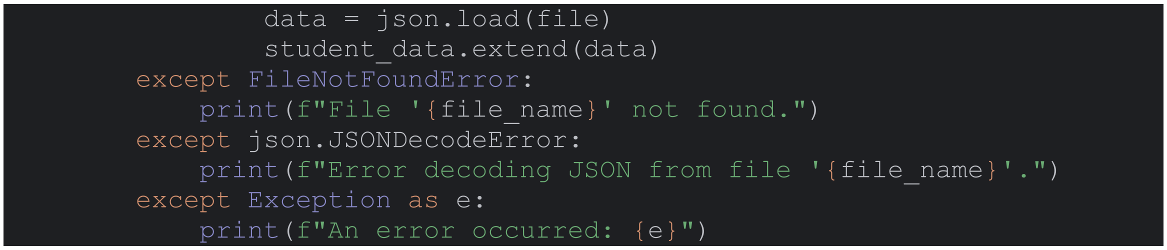
```
@staticmethod
def display_output(message):
    """Outputs the program menu."""
    print(message)
```

Parameters

The next steps was to tell the program to Read Data from File. Implement the read_data_from_file method in the FileProcessor class. This method reads data from a file specified by the file_name parameter and appends it to the student_data list. Use a try-except block to catch any exceptions that may occur during file reading and call the output_error_messages method from the IO class to display error messages.



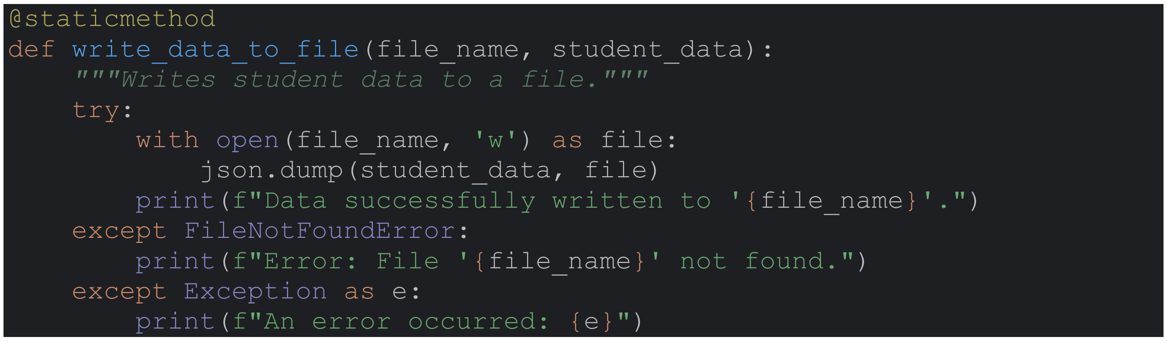
```
@staticmethod
    def read_data_from_file(file_name, student_data):
        """Reads data from a file into the student data list."""
        try:
            with open(file_name, 'r') as file:
```



```
                data = json.load(file)
                student_data.extend(data)
        except FileNotFoundError:
            print(f"File '{file_name}' not found.")
        except json.JSONDecodeError:
            print(f"Error decoding JSON from file '{file_name}'.")
        except Exception as e:
            print(f"An error occurred: {e}")
```

Data to File

Then I had to implement write_data_to_file and use a try-except block to catch exceptions.



```
@staticmethod
def write_data_to_file(file_name, student_data):
    """Writes student data to a file."""
    try:
        with open(file_name, 'w') as file:
            json.dump(student_data, file)
        print(f"Data successfully written to '{file_name}'.")
    except FileNotFoundError:
        print(f"Error: File '{file_name}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")
```

Input/Output

I had to implement the remaining methods in the IO class for handling input/output operations and methods include output_error_messages for displaying error messages, output_menu for displaying the program menu, input_menu_choice for prompting the user for a menu choice, input_student_data for prompting the user for student data and adding it to the list, and output_student_courses for displaying student course information. I also used error handling mechanisms such as try-except blocks to handle invalid inputs and exceptions.

Summary

Module 06 covered Functions. We learned of functions, parameters, and arguments as well as their similarities and usages. Lab 1 was on Functions, Lab 2 was on Parameters, and Lab 3 was Working with Classes and SoC. We finished the module with assignment 06 and posting our code onto Github.