Jenna Ho
Nov. 19, 2023
Foundations of Programming: Python
Assignment06
https://github.com/jennah2001/IntroToProg-Python-Mod06

Module06 Completing the Assignment

Introduction

In Module06, the use of functions, classes and the separation of concerns pattern are introduced. Classes can be used to describe objects, and functions are used to perform operations on data described. By using functions and classes, we can shorten our code, organize and assign our data better. JSON files are also used in this assignment, which was introduced in the previous lecture. In addition, Github is being used to store the codes we have written as well as this document.

Data Constants

First import file is stated at the top of the file. Then the data constants are stated. The data variables stayed the same as past assignments, this includes the student's first name, last name, course name, menu choice and more.

```
# ------------------------------------------------------------------------------------ #
import json

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

# Define the Data Variables and constants
student_first_name: str = ''  # Holds the first name of a student entered by the user.
student_last_name: str = ''  # Holds the last name of a student entered by the user.
course_name: str = ''  # Holds the name of a course entered by the user.
student_data: dict = {}  # one row of student data
students: list = []  # a table of student data
csv_data: str = ''  # Holds combined string data separated by a comma.
json_data: str = ''  # Holds combined string data in a json format.
file = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.
```

Figure1: Data constants

Define Classes

For this assignment, there's a requirement of using classes. The program should have one class named FIleProcessor, and one class named IO. It is important to note that all classes need to have descriptive document strings. There are also a number of functions required for the assignment, this includes output error messages, output menu, input menu choice, output student courses, input student data, read data from file, as well as write data to file.

For Class IO, it is a class for all inputs and outputs of student data. I used @staticmethod decorator to have a function pass as a parameter. Then I used def to define the output error messages as a function, and printed out the error messages. I repeat the use of @staticmethod decorator, and def to print out menu and menu choice. Then I added return menu choice to return back to menu choice. I also repeated the same steps for output student courses, which add students into the student list. Same steps for defining input student data as well, where it adds students into student data.

```
class IO:
    '''
    print out error message
    '''
    4 usages (2 dynamic)
    @staticmethod
    def output_error_messages(message: str, error: Exception = None):
        print(message)
        print(error.__doc__)
        print(error.__str__())
    '''
    print out menu
    '''
    1 usage
    @staticmethod
    def output_menu(menu: str):
        print(menu)
    '''
    print out menu choice
    '''
    1 usage
    @staticmethod
    def input_menu_choice():
        menu_choice = input("What would you like to do: ")
        return menu_choice
    # return back to where it is called, and input in variable:menu choice
    '''
    taking entire student data list and printing it out
    '''
    2 usages (1 dynamic)
    @staticmethod
    def output_student_courses(student_data: list):
        for student in student_data:
```

Figure 2: classes and Functions Part 1

For class FileProcessor, it is a class that reads the file, inputs the student data, and writes student data to the file. I defined __init__ as the constructor for the class IO,a dn the self parameter refers to the object io. This is to call the instance variable within its own class. I used https://www.geeksforgeeks.org/__init__-in-python/ as the outside resource to understand this concept. Then to read front the file and put it in the student list, I again used @staticmethod decorator and def to open and read the file, then JSON file is loaded, then file will close. I also used Exception to catch any error, and print out the message that there is an error. Then I used finally to close the file if it is not closed. I repeated the same steps for writing data to a file,

where I open the file and write to it, then JSON answers and writes the student to the file. I then again used Exception to catch any error and close the file.

```
85  class FileProcessor:
86      # instance variables: properties of the class
87      def __init__(self):
88          self.io = IO()
89
90      # read from the file and put it in the student list
        1 usage
91      @staticmethod
92      def read_data_from_file(self, file_name: str, data: list):
93          try:
94              file = open(file_name, "r")
95
96              # JSON Answer
97              data = json.load(file)
98
99              file.close()
100             return data
101         except Exception as e:
102             self.io.output_error_messages("there is a problem with reading the file", e)
103             # e.doc---explain the reason of the mistake/ str--convert exception into readable format
104         finally:
105             if file.closed == False:
106                 file.close()
107     '''
108     write student list to the file
109     added self to call the instance variable within its own class
110     '''
        1 usage
111     @staticmethod
112     def write_data_to_file(self, file_name: str, student_data: list):
113         try:
114             file = open(file_name, "w")
115
116             # # JSON answer-- writes the student to the file
```

Figure 3: classes and Functions Part 2

Options

When the program starts, we want the file data to be read into a list of lists. To call the read data from the file function, I defined the fileprocessor as well as students. To be able to read class objects and call it, I defined io. The next step is to present and process the data.

Menu options are presented using true statement, and loops and commands are used for each option. I call out the menu using functions. For option 1, student's first name, last name, and course name need to be stored from the inputs. The first names and last names must be

alphabetical, so an if not statement is written to detect any input that's not alphabetical. Then the program can proceed once the correct information is inputted, and the data is appended.

```
fileprocessor = FileProcessor()
students = fileprocessor.read_data_from_file(fileprocessor, file_name=FILE_NAME, data=students)
# make the class object and be able to call it
io = IO()
# Present and Process the data
while (True):

    # Present the menu of choices-- Call out menu using function2
    io.output_menu(MENU)
    menu_choice = io.input_menu_choice()

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!

        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The last 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_data = {"FirstName": student_first_name,
                            "LastName": student_last_name,
                            "CourseName": course_name}
            # students.append(student_data)
            io.input_student_data(students, student_data)
            print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
        except ValueError as e:
            io.output_error_messages( message: 'there are numbers in your input', e)
        except Exception as e:
            io.output_error_messages( message: 'There was a problem with your entered data', e)
        continue
```

Figure 4: Writing Options

For option 2, I didn’t change anything from the last assignment. For option 3, I wrote a line of code to save the data to a file using write to file. For option 4, I close the program. It remained the same as last assignment.

Results

My code runs successfully in both PyCharm and the terminal. I am able to write inputs when choosing option1, and the JSON file data as well as my input from option 1 is presented when choosing option2. Option 3 saves the data, and option 4 quits the program.

```
What would you like to do: 1
Enter the student's first name: Jenna
Enter the student's last name: Ho
Please enter the name of the course: Python 100
You have registered Jenna Ho for Python 100.

---- 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.
----------------------------------------

What would you like to do: 2
--------------------------------------------------
Student Bob Smith is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student jenna ho is enrolled in pythn100
Student Jenna Ho is enrolled in Python 100
--------------------------------------------------

---- 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.
----------------------------------------

What would you like to do: 3
The following data was saved to file!
```

Figure 5: Results Part 1

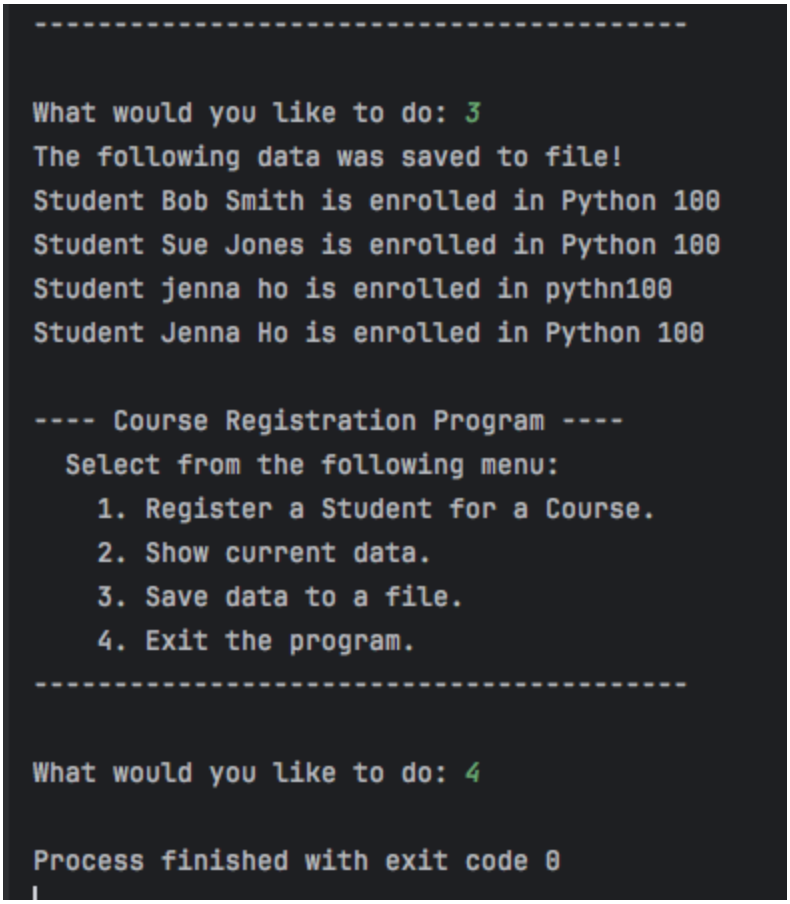

Figure 6: Results Part 2

Summary

In module06 assignment, I am able to include the use of functions and classes successfully. I utilized class lectures, outside resources, as well as the demos from the module file to complete this assignment.