Name: Diellza Mehmeti
Date: 11/27/2024
Course: Foundations of Programming: Python
Assignment: 07
https://github.com/diellzak1/IntroToProg-Python-Mod07

Assignment 07

Introduction

In this week, I continued to work on classes and function and a few new things that were introduced. In this module we created two classes, Student and Person. I used the getter and setter functions within these two classes to access and modify attributes. This is where we also did the error handling. I ensured that the rest of the classes and functions were updated so that those refer to the Student/Person class as appropriate. I ran a few testing in Python and CMD to see if everything works as intended. Finally, I created a repository in GitHub where I saved the script and the documentation.

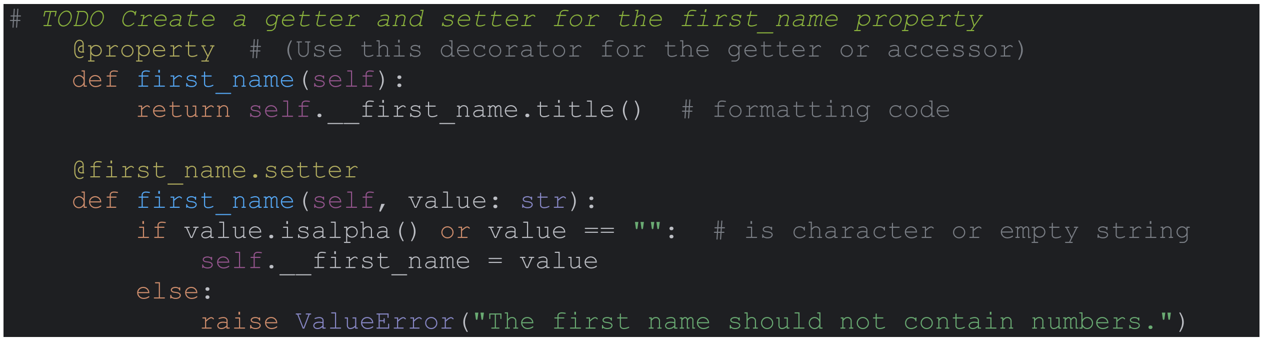
Script Header, Comments, Constants, and Variables

I started the assignment by using the Assignment07 – Starter file to keep the coding standard. I updated the name, script header, the descriptions for the classes and functions. Comments were set and I added a few to document my code. I then declared the two constants given in the assignment and two variables. I moved the two variables before the main body of the code.

Classes and Functions

Besides the FileProcessor and IO class, this week, I created two other classes: Student and Person which would represent the data on student first name, last name, and course name. In these two new classes, I added the getter and setter function to access and modify attributes. I ensured that the error handling was removed from the other IO class and instead I added it in the Person class. There was no need to add error handling in the Person class since we did not need to add any for the course name. I ensured to update the FileProcessor class by creating a new variable, file_data, that is used for reading and writing into the json file. I tested the errors in this part of the code, and they worked as intended.

Here is the example of the Getter/Setter in PyCharm:

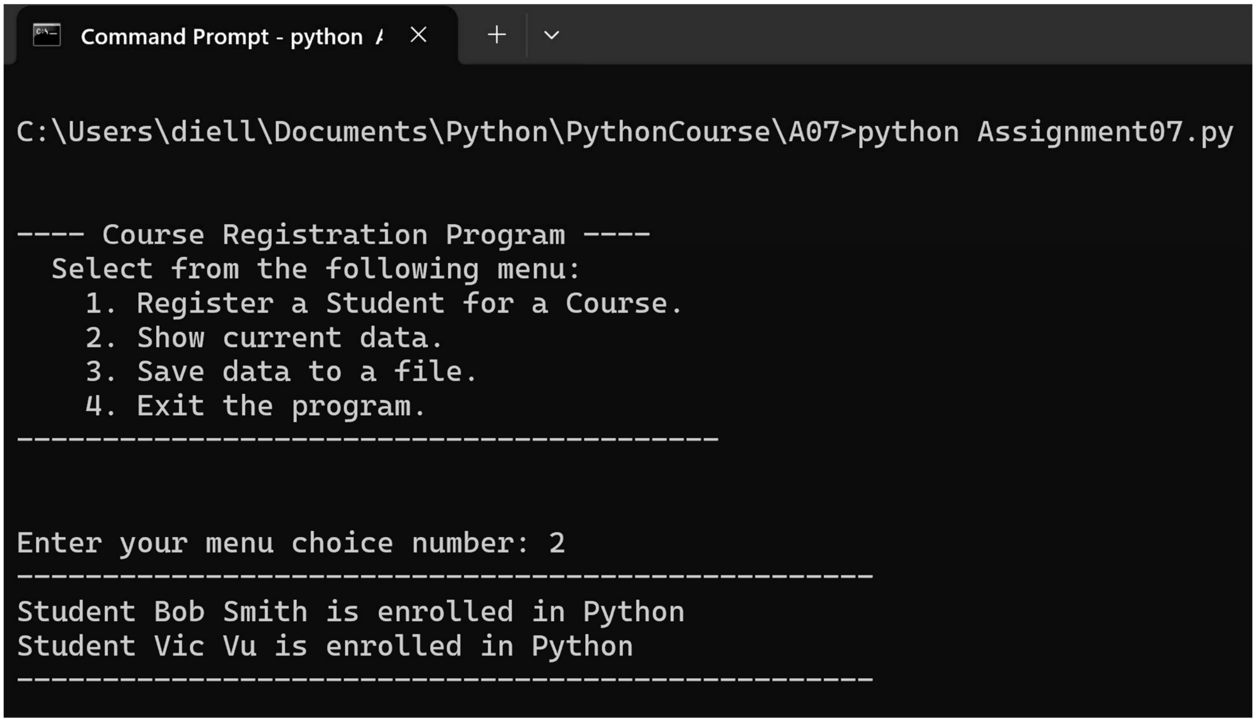


```
# TODO Create a getter and setter for the first_name property
    @property  # (Use this decorator for the getter or accessor)
    def first_name(self):
        return self.__first_name.title()  # formatting code

    @first_name.setter
    def first_name(self, value: str):
        if value.isalpha() or value == "":  # is character or empty string
            self.__first_name = value
        else:
            raise ValueError("The first name should not contain numbers.")
```

Testing and running the script

I tested the code after each change that I made by using the breakpoints. As the code is getting more complex, I found it helpful to do continues testing. I ran in some errors when updating the list by not using the same key as found in the file. This was quickly fixed by matching the names in the Enrollment file and between read and write functions. I also tested scenarios if the first name and last name contain numerical values, and this worked as intended. Once I finalized the testing portion, I ran the code in PyCharm, and then in CMD. I saved the file with the "py" extension, uploaded it in the GitHub and shared it in the discussion board for my peers to review.



```
C:\Users\diell\Documents\Python\PythonCourse\A07>python Assignment07.py


---- 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: 2
--------------------------------------------------
Student Bob Smith is enrolled in Python
Student Vic Vu is enrolled in Python
--------------------------------------------------
```

Figure 1: Running Python code in CMD and displaying menu choice 2

Summary

In this week's assignment, I continued to work from assignment six and reorganized the starter code. This was another challenge module where I learned about constructors, attributes, and properties. I created two new classes in which I used the constructor to set attributes. I also used two new functions, getter and setter which make it easier to access data and format it. I worked slowly through the code by ensuring frequent testing and error handling. Once the code was tested, I ran it in PyCharm and CMD. Finally, I saved the code in GitHub and shared it with my peers for review.