Name: Phuong Nguyen
Date: 6/5/2025
Course: Foundations Of Programming: Python
GitHub: https://github.com/tnyen102/IntroToProg-Python-Mod07

Assignment 07 – Classes and Objects

Introduction

This week, I learned about classes and objects and how they are inter-related. We can create object instances from a class if we want to access unique data to it. I also learned about the different classes and their purposes (processing, presentation, and data). I also learned the special method of constructor that automatically called an object of a class. An important keyword in this module is "self" that is used to refer to data or functions found in an object instance. Lastly I learned about Git vs GitHub and how it can integrate with PyCharm.

Writing the Script

This week's assignment is similar to Assignment 06 except we have to sets of data classes. The basic information was already given with the starter file, so I copied over the header, the constants and variables over and updated it with what is required for this assignment. In addition I have import the json file over and used the Enrollments.json that was already given (Figure 1).

Assignment07.py

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment07
# Desc: This assignment demonstrates using data classes
# with structured error handling
# Change Log: (Who, When, What)
#   PNguyen,6/4/2025,Created Script
#   <Your Name Here>,<Date>,<Activity>
# ------------------------------------------------------------------------------------------ #
import json

# Define the Data Constants
MENU: str = ("---- Course Registration Program ----.\nSelect from the following menu:.\n1. Register a Student for a Course..\n"
             "2. Show current data.\n3. Save data to a file.\n4. Exit the program.\n---------------------------------------- ")

FILE_NAME: str = "Enrollments.json"

# Define the Data Variables
students: list = []  # a table of student data
menu_choice: str  # Hold the choice made by the user.
```

Figure 1: Basic Information

My next task is similar to create a Person class to represent the person's data. I had to create functions to define each data fields which are first name and last name (Figure 2).

```python
class Person: # TODO Create a Person Class  1 usage
    """class to represent person data, first_name (str): student's first name, last_name (str): student's last name"""

    # TODO Add first_name and last_name properties to the constructor
    def __init__(self, first_name: str = "", last_name: str = ""):
        self.__first_name = first_name
        self.last_name = last_name

    # TODO Create a getter and setter for the first_name property
    @property  15 usages (12 dynamic)
    def first_name(self):
        return self.__first_name.title()

    @first_name.setter  13 usages (12 dynamic)
    def first_name(self, value: str):
        if value.isalpha() or value == "":
            self.__first_name = value
        else:
            raise ValueError("The first name should not contain numbers.")

    # TODO Create a getter and setter for the last_name property
    @property  14 usages (12 dynamic)
    def last_name(self):
        return self.__last_name.title()

    @last_name.setter  13 usages (12 dynamic)
    def last_name(self, value: str):
        if value.isalpha() or value == "":
            self.__last_name = value
        else:
            raise  ValueError("The last name should not contain numbers.")

    # TODO Override the __str__() method to return Person data
    def __str__(self):
        return f"{self.first_name},{self.last_name}"
```

Figure 2: Person Class

The next is Student class which inherits from the Person class. This class is to represent the student data which includes the first name, last name, and course information. I also had to create different functions including the add getter and setter for the course name and return the value with all 3 information. (Figure 3).

```python
class Student(Person): # TODO Create a Student class the inherits from the Person class
    """Class representing student data"""
    def __init__(self, first_name: str = "", last_name: str = "", course_name: str = ""):

    # TODO call to the Person constructor and pass it the first_name and last_name data
        super().__init__(first_name=first_name, last_name=last_name)
    # TODO add a assignment to the course_name property using the course_name parameter
        self.course_name = course_name

    # TODO add the getter for course_name
    @property  4 usages (2 dynamic)
    def course_name(self):
        return self.__course_name.title()

    # TODO add the setter for course_name
    @course_name.setter  3 usages (2 dynamic)
    def course_name(self, value: str):
        self.__course_name = value

    # TODO Override the __str__() method to return the Student data
    def __str__(self):
        return f"{self.first_name},{self.last_name},{self.course_name}"


```

Figure 3: Student Class

The next class is FileProcessor to work with the JSON file. This includes reading and saving to the JSON file. The lass class is the Presentation class which is to present the data to the users to get their inputs as well as displaying it back to them. The function is really similar to assignment 6, the main diference is to convert dictionary data to Student data to be able to write to the JSON file. (Figure 4).

```python
            # TODO replace this line of code to convert dictionary data to Student data
            student_objects = []
            for row in json_students:
                student = Student(first_name=row["FirstName"],
                                  last_name=row["LastName"],
                                  course_name=row["CourseName"])
                student_objects.append(student)

        except Exception as e:
            IO.output_error_messages(message="Error: There was a problem with reading the file.", error=e)

        finally:
            if file.closed == False:
                file.close()

        return student_objects

```

```
            # TODO Add code to convert Student objects into dictionaries (Done)
            json_students = []
            for student in student_data:
                student = {"FirstName": student.first_name,
                           "LastName": student.last_name,
                           "CourseName": student.course_name}
                json_students.append(student)

            file = open(file_name, "w")
            json.dump(json_students, file)
            file.close()
            IO.output_student_and_course_names(student_data=student_data)
        except Exception as e:
            message = "Error: There was a problem with writing to the file.\n"
            message += "Please check that the file is not open by another program."
            IO.output_error_messages(message=message,error=e)
        finally:
            if file.closed == False:
                file.close()
```

Figure 4 Convert To JSON

The last script remains the same as in assignment 6. This loop allows the users to make the selection and the output will be what is selected. (Figure 5).

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

# Present and Process the data
while True:

    # Present the menu of choices
    IO.output_menu(menu=MENU)

    menu_choice = IO.input_menu_choice()

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        students = IO.input_student_data(student_data=students)
        continue

    # Present the current data
    elif menu_choice == "2":
        IO.output_student_and_course_names(students)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
        continue

    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, or 3")

print("Program Ended")
```

Figure 5 While Loop with the functions

Testing the Script

With my script in place, it is time for me to test and make sure it works in both PyCharm and the Command Prompt. After some trial and error, I was able to successfully get my script to work in PyCharm, it also opened and saved the json file as intended (Figure 6). It also worked in the command prompt, as you can see the json file was updated and saved to a later time when I ran it (Figure 7). The error message also works as intended (Figure 8).

```
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: 1
Enter the student's first name: Phuong
Enter the student's last name: Nguyen
Please enter the name of the course: Python 100

You have registered Phuong Nguyen 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.
----------------------------------------

Enter your menu choice number: 2
------------------------------------------------
Student Vic Vu is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Phuong Nguyen 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.
----------------------------------------
```

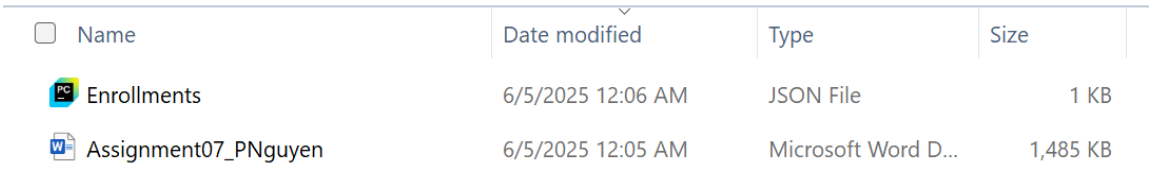
```
Enter your menu choice number: 3
-------------------------------------------------
Student Vic Vu is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Phuong Nguyen is enrolled in Python 100
-------------------------------------------------

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

Enter your menu choice number: 4
Program Ended

Process finished with exit code 0
```

```
, {"FirstName": "Phuong", "LastName": "Nguyen", "CourseName": "Python 100"}]
```



Name	Date modified	Type	Size
Enrollments	6/5/2025 12:06 AM	JSON File	1 KB
Assignment07_PNguyen	6/5/2025 12:05 AM	Microsoft Word D...	1,485 KB

Figure 6: Test in PyCharm

```
---- 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: 1
Enter the student's first name: Test
Enter the student's last name: Demo
Please enter the name of the course: Python 100

You have registered Test Demo 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.
----------------------------------------

Enter your menu choice number: 2
------------------------------------------------
Student Vic Vu is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Phuong Nguyen is enrolled in Python 100
Student Test Demo 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.
-----------------------------------------

Enter your menu choice number: 3
-------------------------------------------------
Student Vic Vu is enrolled in Python 100
Student Sue Jones is enrolled in Python 100
Student Phuong Nguyen is enrolled in Python 100
Student Test Demo 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.
-----------------------------------------

Enter your menu choice number: 4
Program Ended

C:\Documents\Python\PythonCourse>
```

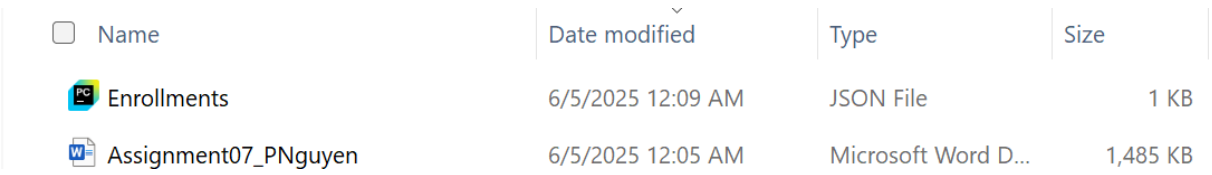


Figure 7: Test in Command Prompt

```
---- 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: 1
Enter the student's first name: 142
One of the values was the correct type of data!

-- Technical Error Message --
The last name should not contain numbers.
Inappropriate argument value (of correct type).
<class 'ValueError'>

---- 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: 1
Enter the student's first name: Tom
Enter the student's last name: 324
One of the values was the correct type of data!

-- Technical Error Message --
The last name should not contain numbers.
Inappropriate argument value (of correct type).
<class 'ValueError'>

---- 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.
```

Figure 8: Error Message

Summary

This week's assignment was challenging especially on how to work with JSON files. I had to do some research to figure out why I was getting an error message regarding option 3 why I can't save the data to the JSON file. This module taught me more about classes and objects. I also learned about GIT which will be very useful in the future to write and edit my codes.