

Name: Kamaljeet Gill

Date :05/27/2024

Course: Foundation of Programming Python

Assignment 7: Classes and Objects

GitHub Link: <https://github.com/gillkamal20/IntroToProg-Python-Mod07>

#### Topic covered in Module 7:

- This module started with the concept of classes and Functions. Functions work with data that is passed to them as arguments, while classes manage data using attributes and methods.
- The task of constructors is to initialize (assign value) to the data members of the class when an object of the class is created. In Python the `__init__()` method is called the constructor and is always called when an object is created.
- Python property () is a built-in function for creating and returning a property object. It offers an interface to instance attributes.
- Inheritance in Python allows us to define a class that inherits all the methods and properties from another class. Parent class is the class being inherited from, also called base class. Child Class is the class inherits from another class, also called derived class.

#### Assignment

I started this assignment by creating class for class Person and used a getter and setter for first name and last name of Student.

```
# Creating property for first name
1 usage
@property
def first_name(self) -> str:
    return self._first_name.title()

# Create a getter and setter for the first_name property
@first_name.setter
def first_name(self, value: str):
    if value.isalpha() or value == "":
        self._first_name = value
    else:
        raise ValueError("First Name should not contain numbers")
```

Next thing I learned is to use `_init_()` method to override and add course name to already existing file

```
def __init__(self, first_name='', last_name='', course_name=''):
    super().__init__(first_name, last_name)
    self._course_name = course_name
```

Next Step, I converted the list to Student dictionary and used variable as show in Screen shot: Variable

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

Screen shot: Variable

I changed the read\_data\_from\_file to above variable and faced error as shown in Screen Shot:Error

[illegible]

### Screen Shot: Error

I fixed the error by adjusting the name in enrollment.json file. It was FirstName, LastName and CourseName. I changed it to sync with name used in Python file.

Output

When Option 1 is selected

```
Enter your menu choice number: 1
Enter the student's first name: kamal
Enter the student's last name: Jeet
Please enter the name of the course: SQL

You have registered kamal Jeet for SQL.
```

For Option 2, I faced the below error:

```
-----
Traceback (most recent call last):
  File "C:\Python\A07\Assignment07.py", line 286, in <module>
    IO.output_student_and_course_names(students)
  File "C:\Python\A07\Assignment07.py", line 227, in output_student_and_course_names
    print(f'Student {student["FirstName"]} '
                        ^^^^^^^^^^^^^^^^^^
TypeError: 'Student' object is not subscriptable
```

In order to solve this, I changed the code as below:

```
@staticmethod
def output_student_and_course_names(student_data: list[Student]):
    """ This function displays the student and course names to the user

    ChangeLog: (Who, When, What)
    RRoot,1.1.2030,Created function

    :param student_data: list of dictionary rows to be displayed

    :return: None
    """

    print("-" * 50)
    for student in student_data:
        print(f'Student {student.first_name} '
              f'{student.last_name} is enrolled in {student.course_name}')
    print("-" * 50)
```

After correction Output for Option 2 is as follow

Option 2 Result:

```
Enter your menu choice number: 2
-----
Student Bob Smith is enrolled in Python 100
Student Ric Roy is enrolled in Python
Student Amy Vu is enrolled in SQL
-----
```

Option 3 On Choosing option 3, below error occurred

```
Enter your menu choice number: 3
Error: There was a problem with writing to the file.
Please check that the file is not open by another program.

-- Technical Error Message --
Object of type Student is not JSON serializable
Inappropriate argument type.
<class 'TypeError'>
```

I missed updating the list in json.dump () function, fixed this error by updating the code as

```
file = None

try:
    file = open(file_name, "w")
    json.dump(file_data, file)
    file.close()
    IO.output_student_and_course_names(student_data=student_data)
```

Output for option 3 after correction:

```
Enter your menu choice number: 3
-----
Student Bob Smith is enrolled in Python 100
Student Ric Roy is enrolled in Python
Student Amy Vu is enrolled in SQL
-----
```

### Summary

While testing first name in setter, I was able to enter number Example: when I entered John1, no error message prompted and John1 was printed. I will continue exploring more on error handling. Similar point was also raised in today`s class for Lab 3.