



Python OOP – Object Oriented Programming for Beginners

Mini Project

Methods

Project Description:

A **music school** has hired you to **add functionality** to an existing program that is used to keep a centralized record of all their students across their headquarters.



- Your job is to **add three methods** to the existing program:

1. A method `print_students_data` that prints the name of each one of the students in the dictionary, their age, and the classes they are taking, one student per line.

A sample output would be:

```
"Student: Gino who is 15 years old and is taking ['Piano', 'Guitar']"  
"Student: Talina who is 28 years old and is taking ['Chello']"  
"Student: Eric who is 12 years old and is taking ['Singing']"
```

2. A method `print_student` that prints the string shown above with the name, age, and classes of a student. It should take the name of the student as an argument and print only the data of that particular student. (Tip: you might consider using this method in the `printStudentsData` method to avoid repetition).

3. A method `add_student` that adds a student to the existing `students` dictionary if the school is not full. If it is full, it should print a message. The key used to add the student to the dictionary should be the name of the student and the value should be a list with the age, phone number, and classes that the student is taking. The method should take the `name` of the student as a parameter and a list with the `data` associated with that name as another parameter.

After adding these methods, you should create an **instance** of `MusicSchool` with 56 as the maximum number of students, 8 working hours, and 15000 as the initial revenue. Then, call each method through that instance like this:

```
<instance>.print_students_data()
<instance>.print_student("Gino")
<instance>.add_student("Jack", [60, "562-234-234", ["Piano"]])
```

Note: remember that `students` is a class attribute. You will need to use a particular syntax to access it in your methods.

Challenge: Right now, the existing program does not store the data in a file, so new student records are lost when the program ends. Could you make this program save the students' information to a `.txt` file using methods? This is not required to submit the assignment, but it's an interesting program that you might enjoy working on.

This is the existing program:

```
class MusicSchool(object):

    students = {"Gino": [15, "653-235-345", ["Piano", "Guitar"]],
                "Talina": [28, "555-765-452", ["Chello"]],
                "Eric": [12, "583-356-223", ["Singing"]]}

    def __init__(self, max_num_students, working_hours, revenue):
        self.max_num_students = max_num_students
        self.working_hours = working_hours
        self.revenue = revenue

    # Add your methods below this line


# Create the instance


# Call the methods
```

Tips:

This is a quick refresher of Python dictionaries. Their elements are key-value pairs. Each key corresponds to a value: {"key1": "value1", "key2": "value2"}. Key-value pairs are separated using commas.

This is the basic syntax to work with dictionaries:

- To iterate over the keys in a dictionary: `for key in <dictionary>:` (key can be any name but the values assigned will be the keys of the dictionary).

- To iterate over the keys and their corresponding values in a dictionary simultaneously: `for key, value in <dictionary>.items():`

- To access a value in a dictionary, you use its corresponding key:
`<dictionary>[<key>]`

- To print data stored in a dictionary, you can use `print()` and include the expression (converted to a string) or string within parenthesis.

For example: `print(<dictionary>[<key>])`

Note: `<dictionary>` represents a variable that references a dictionary.

Note: You will be able to check your solution with a sample answer as soon as you submit your mini project.