Question Text:

Test Name: Xplore_OPA_15May2020_Python

Qn : OPA-Python-Python-University Management

Title: University Management

Create a class **Professor** with the below attributes

profId of integer type
profName of string type
subjectsDict is of dictionary of subjects{subject is the Key :years of experience as a value}

where , profId represents professor id, profName represents professor name and subjectsDict represents the dictionary with Subject Name , Number of years of experience as Key:Value pair.

Define the **__init__** method to initialize the attributes in the above sequence.

Note: *The dictionary is created and filled in main program for each of the Professor after reading the professor related data and passed as the third argument to this constructor and this will be initialized to subjectsDict.*

Create a class **University** with the following methods

Define two methods – **getTotalExperience** and **selectSeniorProfessorBySubject** in University class as per the below description.

getTotalExperience method:
This method takes two input parameters -the list of Professor objects and a number value which represents the professor id. It returns the total experience of that professor whose id matches with the given professor ID.
Total experience is to be calculated as the sum of experience of all the subjects of the respective professor. The experience of each subject can be accessed from the dictionary belonging to the professor object.
If no professor is available with the given id, then return 0

Hint:
a. The Professor list needs to be created and filled in main program after reading each professor related data and passed as the first argument to this constructor and this will be used to traverse the respective professor object to get the subject wise experience to calculate the total experience.
b. Display the total experience (returned by this function) in the main function.

selectSeniorProfessorBySubject method:

This method takes two parameters: the list of professor objects and a string representing the subject name. The method returns the professor who has the highest experience in that subject(Assume no professor have same years of experience in a given subject).

If there is no professor available for a given subject, then return “None“

***Note**: All string searches are to be case insensitive.*

Hint:
a. Use the same list which is supplied as argument to the method, getTotalExperience.
b.Display the Professor object(returned by this function) in the main function at last

e.g. Output Format to display the return value of the above method is as below
1 Shivakumar {'Maths': 10, 'Physics': 10, 'Chemistry': 10}
Where in '1' represents the Professor Id, 'Shivakumar' is the Professor Name
and '{'Maths': 10, 'Physics': 10, 'Chemistry': 10} ' represents
the dictionary of the subjects and Experience as key:value pairs of the professor ,whose is senior in the given subject

Note:
a. You would required to write the main program completely, please follow the below instructions for the same a.
b. Your main should read the sample input data as mentioned below and create the respective objects and call the methods(getTotalExperience and selectSeniorProfessorBySubject) mentioned above in the same order as they appear in the question text from main function.
Refer the comments in the default code editor below , which provides more clarity to implement the code.

Sample Input (below) description:

1. The 1st input taken in the main section is the number of Professor objects to be added to the list of Professors.

2. The next set of inputs are for a professor object
   i.e. professor id,professor name,number of subjects.
   For each of the subject,
   read subject name and experience in years in that subject and add it to the respective subjects dictionary of the professor as key:value pair

3. The above point (Point#2) is repeated for each and every professor for the number of Professor objects given in the first line of input.

4. The last 2 lines of the sample input represents the professor id for finding the total experience and subject

Sample Input For Custom Input Testing

```
4
1
Shivakumar
3
```

```
Maths
10
Physics
10
Chemistry
10
2
Rajesh
4
MATHS
5
PHYSICS
5
CHEMISTRY
5
COMPUTERS
5
3
Vasudev
2
MATHS
4
PHYSICS
4
4
Srinivas
3
Maths
8
Physics
8
Chemistry
8
3
maths
```

Sample Output

```
8
```

```
1 Shivakumar {'Maths': 10, 'Physics': 10, 'Chemistry': 10}
```

Referring to the above sample input data we see that, Professor with id 3(Professor Name : Vasudev) teaches the following subjects:

Maths with experience of 4 years
Physics with experience of 4 years.

Hence, total experience = 4+4 =8

Also, In the given list, we see that each professor teaching Maths has the following experience:
Shivakumar with experience of 10 years
Rajesh with experience of 5 years
Vasudev with experience of 4 years
Srinivas with experience of 8 years.

Hence, professor with highest experience of teaching Maths is Shivakumar, Hence displayed the Shivakumar's Object in the format mentioned in the output.

Coding Editor :

Comments to implement the solution:

 # Enter your code here. Read input from STDIN. Print output to STDOUT

#Define Professor class - after this line

#Define University Class - after this line

"""Define main function below
1. Read number of professors you want to consider for the problem statement
2. For each of the professor perform below actions
a. Read Professor related data - Id , Name and number of subjects
a1. For each of the Subject,
-Read the first subject name and years of experience in the respective subject
and add the first subject name and experience in years as key:value pair
to the subjects dictionary (as a first element) of the first professor

-Similarly read the second subject name and experience and add this second subject

details to subject dictionary(Subject name as a Key and 'Experience in Years' as value).

Reading the subject related data and adding the data to subjects dictionary is repeated for the number of subjects, read in the point #a

a2. Create a professor object with the data (read above) inlign to the constructor mentionedin question text and add it to list of professors

b. Repeat step#a for the number of times(number of professors to be added to professor list) referred in the point#1 and read different professors data in each iteration as menioned in point#a and append to professor list

3. Read Professor Id , to pass to getTotalExperience method as second argument to find the total experience

4. Read SubjectName, to pass to selectSeniorProfessorBySubject method as second argument to find the professor who has more experience in the subject

5. Create University object and call the functions mentioned in the Qn text

6. Finally display the ouput as mentioned in the Question text."""