Test Title: Xplore_OPA_22May2020_Python
Question Title : OPA-Python: Hospital management

Create a class **Doctor** with below attributes:
doctorId – Numeric type
doctorName – String type
specialization – String type
consultationFee – Numeric type

doctorId represents the unique identification of a Doctor object,
doctorName represents the name of the doctor,
specialization represents the doctors specialization and consultationFee represents the doctor fee.

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

Create a class **Hospital** with below attributes:
doctorDB - is of type dictionary with Doctor objects [**Serial number of a Doctor** in the Hospital and **the respective Doctor object** as **key** : **value** pair]
doctorName_searchFor - is of string type

Define the __init__ method to initialize the attributes in the above sequence . It initializes the dictionary of Doctor objects with the dictionary supplied from main program while creating the **Hospital** object.

Note: The dictionary is created and filled in main program by adding each Doctor object, which is created with the input data related to a respective Doctor and passed as the first argument to this constructor and this will be initialized to doctorDB

Define two methods
– **searchByDoctorName** and **calculateConsultationFeeBySpecialization** in **Hospital** class.

searchByDoctorName:
This method will find the respective **Doctor** object based on the doctor name and return a **list** of Doctor objects to main program, whose name if matches with the given name ,supplied as an argument.

If there is no Doctor found with the given name then return NULL object , ie None in python

Hint:
a. Use the dictionary, doctorDB in Hospital object to find out the Doctor object(s) in the dictionary of Doctor objects, based on the given Doctor name.
b. Display the Doctor object (returned by this function) in the main function

calculateConsultationFeeBySpecialization:
This method will take a **Doctor specialization** as parameter and return the total **consultationFee** of all the Doctors ,whose specialization is same as supplied as an argument from the main program. If there is no Doctor found with the given specialization then return 0 to main program.

These methods should be called from the main function / program.

Hint
a. Use the dictionary, doctorDB in hospital to get the consultation fee of each of Doctor (Doctor object in the Hospital) for the given specialization supplied as argument .
b.Display the Total Fee in the in the main function

Note:
a. You would required to write the main program completely, Please follow the below instructions for the same.
b. You would required to write the main program which is inlign to the s**ample input data section** mentioned below and **to read the same** .
c. Create the respective objects(Doctor and Hospital)
Doctor object after reading the data related to Doctor and add the doctor object to Doctors dictionary
Hospital Object with the dictionary of Doctor objects, created in the previous step
d. Finally call the methods
(**searchByDoctorName** and **calculateConsultationFeeBySpecialization)** mentione d above in the same order , they appear in the question text from main function .
e. Display the data returned by the functions , in the main program as per the format mentioned in the sample output.
If no Doctor exists in with the given name then display the message "No Doctor Exists with the given DoctorName" in the main function.
If no Doctor exists with the given specialization then display the message "No Doctor with the given specialization" in the main function.

Sample Input (below) data description:
1.The 1st input taken in the main section is the number of Doctor objects to be created and added to the dictionary of Doctors in the Main program
2.The next set of inputs are
the doctorId, doctorName, specialization and consultationFee of first Doctor
3. For each Doctor object repeat point#2 and this point is repeated for number of Doctor objects given in the first line of input
4.The last but one line of input refers the doctorName to be searched ie an argument for **searchByDoctorName** function
5. Last line of input represents the specialization, supplied as an argument to **calculateConsultationFeeBySpecialization** function, to get the total **consultationFee** of all the Doctors for a given specialization.

Sample Input :

```
4
90901
GovindRajulu
ENT
500
90902
Madhuri
Dermatologist
700
90903
Divya
Gynaecologist
600
90904
Suryam
Cardiologist
900
Madhuri
Cardiologist
```

Output :

```
90902
Madhuri
Dermatologist
700
900
```