CSE201 Advanced Programming

Lab Assignment-01

IIIT-Delhi. 4th Sept 2020. Due by 11:59 pm on 6th Sept 2020

This assignment is a take-home lab assignment. No extensions whatsoever will be provided. Any submission after the deadline will not be evaluated. If there is any ambiguity or inconsistency in a question, please seek clarification from the teaching staff. Please read the entire text below very carefully before starting its implementation.

Plagiarism: All submitted lab assignments are expected to be the result of **your individual effort.** You should never misrepresent someone else's work as your own. In case any plagiarism case is detected, it will be dealt as per IIITD plagiarism policy and without any relaxations:

https://www.iiitd.ac.in/sites/default/files/docs/education/AcademicDishonesty.pdf   Please   note that you are not allowed to discuss the lab assignment's design/solution (e.g. classroom page discussions, etc.). Anyone who is found doing this will be treated as a plagiarism case. No excuses!

NOTE that we won't respond to any query that is because you missed the read the description carefully. We will ONLY respond to valid questions. Make sure you ask all your doubts in advance and not at the last minute.

Based on Lectures 01-03

Delhi Government has approached our institute IIIT-D to design a software application to simplify the procedure of admitting patients identified to be infected by CoronaVirus to various

Health Care Institutes. The health coordinator at a particular camp will use your software to onboard newly identified patients and allocate them to various healthcare institutes that have available beds with them. Since this has been asked in short notice, a command-line interface would do the job. You **must** use Object-Oriented programming concepts taught in the Lectures 1-3 for your software implementation. Detailed description of this software is as mentioned below:

Each patient can register themselves only once. They must be registered with name, age, oxygen levels (90 – 100) and body temperature. A unique ID must also be assigned by the application to each patient automatically. Once a patient has been admitted to a health care institute, your software must store the advised number of days for recovery against the patient's ID.

Every health care institute can have different criteria to handle the patients. Due to expertise restriction, certain institutes might not be able to handle critically ill patients. Hence these criteria can be used to shortlist the patients who can be admitted to particular institutes.

There can be two cases:

1) Firstly fill the patients based on the oxygen levels which are accepted by the health care institute.

2) If there are still available beds, fill the patients based on the criteria for body temperature.

Note: They are independent. Check only oxygen level criteria first, then if needed check for body temperature criteria separately.

If any two patients are having the same oxygen or same temperature levels, either of the patients could be admitted.

A Health Care institute should have a status as **OPEN** or **CLOSED**, based on the available beds with them. Once they have admitted enough patients to fill in the vacant beds, their status should change to **CLOSED**.

Once any healthcare institute has filled all its available beds, the healthcare account status is changed to CLOSED. Later, the user must be able to delete the account of the healthcare institute which is closed. Similarly, for a patient, once they are admitted to an institute, they should not be re-admitted anywhere else by mistake. The application user must be able to delete the admitted patient account too at a later point in time.

Your application must exit once all the patients have been admitted.
Following are the input that must be handled by your application:

1) Add Health Care Institute (This step must automatically onboard patients based on eligibility criteria given and ask for recovery days for each patient admitted)

Inputs – Name, Criteria, Number of available beds

Subsequent Input - Recovery days for each patient admitted (based on eligibility)

Output - Display details of the institute(Institute Name, Eligibility criteria, Number of available beds, Admission Status- Open/Closed,Recovery days for admitted patients)

2) **Remove the accounts of admitted patients**

Output: Display ID no's of patients whose accounts were removed

3) Remove the accounts of health care institutes whose applications are closed

Output: Display name of health care institutes whose accounts were removed **4)**

Display number of patients still in the camp (not admitted yet)

Output - <number> patients

5) Display number of health care institutes admitting patients currently (Open)

6) Display details of the Health Care Institute

Input: Health Care Institute name

Output: Health Care Institute Name, Eligibility Criteria, Number of vacant beds, Admission Status (Open/Closed)

7) Display details of particular patient

Input: ID number

Output: ID number, Temperature, Oxygen levels, Admission status, Name of the Health Care institute if admitted already

8) Display all patients

Output: Display all the patient names and ID's generated

9) Display names of patients admitted in an Institute and their recovery time

Input: Institute name

Output: Display the patient names and their recovery time

Note: Kindly follow the input format at all times. Strictly follow an object-oriented programming approach in its implementation as discussed in-class lectures.

Input Format:

The first line will have the number of patients in the health camp, and their details follow.

Name(string) <space> Temperature(float) <space> oxygen levels(integer) <space> age(integer).

(The number of patients and their details is input only once in the beginning of the application, following which only operations follow)

Following this, every input line refers to the query performed and the necessary inputs for that query.

<u>NOTE: The following Sample case is a general guidance of input/ouput flow for the application. All the integer/string in bold are user inputs, the rest are application prompt messages/outputs (which can be customized as you wish).</u>

Sample case:

5 //Number of patients

Ram 98.4 94 25

Sam 100.4 92 55

Jim 104 91 61

Tim 99 93 60

Kim 100 91 48

8 //Query number

 1 Ram

2 Sam

3 Jim

4 Tim
5 Kim

1 //Query number

AIIMS

Temperature Criteria – **102** //This could be considered as the max temperature of patient who could be admitted

Oxygen Levels – **92** //This could be considered the lowest oxygen level till which you could admit

Number of Available beds – **2**

AIIMS

Temperature should be <= 102

Oxygen levels should be >= 92

Number of Available beds – 2

Admission Status – OPEN

Recovery days for admitted patient ID 1 – **7**

Recovery days for admitted patient ID 2 – **14**

5 //Query number

0 institutes are admitting patients currently

4 //Query number

3 patients

6 AIIMS //Query number and Institute name

AIIMS

Temperature should be <= 102

Oxygen levels should be >= 92

Number of Available beds – 0

Admission Status – CLOSED

9 AIIMS //Query number and Institute name
Ram, recovery time is 7 days

Sam, recovery time is 14 days

7 2 //Query number and Patient ID Sam

Temperature is 100.4

Oxygen levels is 92

Admission Status – Admitted

Admitting Institute - AIIMS

2 //Query number

Account ID removed of admitted patients

 1

2

3 //Query number

Accounts removed of Institute whose admission is closed

AIIMS

1 //Query number

DMS

Temperature Criteria – **104**

Oxygen Levels – **91**

Number of Available beds – **3**

DMS

Temperature should be <= 104

Oxygen levels should be >= 91

Number of Available beds – 3

Admission Status – OPEN
Recovery days for admitted patient ID 3 – **10**

Recovery days for admitted patient ID 4 – **8**

Recovery days for admitted patient ID 5 – **12**