The American University in Cairo
Computer Science and Engineering Department

Fundamentals of Computing II
S2021
Assignment 5

Problem Description:

- Create a clinic-appointment system, where patients can book appointments with doctors of the same specialization.
- Patients request appointments basically at a given time slot with any doctor. Doctors are arranged in a logical ring such that if a doctor is not available at a given slot, the patient will be assigned to the next doctor in the list till all doctors are allocated and then goes back to the first doctor. It may be assumed that the patient will always find an available doctor.
- Any doctor can serve any patient if s/he is free at a given time slot. As well, doctors can apologize for a given appointment.
- The system generates some admin reports.
- Appointments can be cancelled and/or modified.

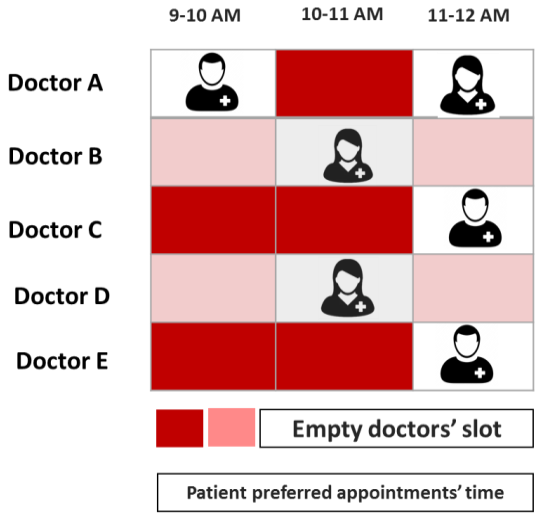
Application's Technical Details:

- Basic Data:
  - The program reads patients' and doctors' info from a file.
  - Each time the program is run patients/doctors' info is loaded automatically in the memory.
  - The file is read line by line and the data is saved in the corresponding variables in the program.

- Appointments' Data Structure:
  - The patient requests an appointment and the system chooses, records and communicates the relevant appointment data.
  - The patients' info is inserted into the doctor's queue at the relevant time slot.
  - There are two kinds appointments:
    - Urgent appointment: Before assigning any appointment for a normal appointment, the urgent appointment should be allocated to any time slot the patient wants.
    - Normal appointment: Each Doctor will be allocated based First Come First Serve.

- It is your choice to implement Doctor, Patient, and Appointment Data Structure as a queue, vectors, or linked list. Yet, the urgent appointment should be implemented as a priority queue i.e max heap.

- An Example of the Appointment allocation will look like this for reserving for 6 patients using patients' preferred timings that will keep the doctors' timing not efficiently allocated but it will definitely please the patients.



Application's Classes:

<u>Class Person</u>:
The class handles generic information about the clinic's stakeholders like doctors and patients. It can be inherited in the future to handle the HR.

Private Variables:

- Name: a structure that includes first name, last name (patient or doctor).
- ID: The National ID of the Person (patient or doctor).
- Any relevant variable(s) you may wish to add

Public Methods:

METHOD NAME	DESCRIPTION
Setters and Getters	Setters and Getters of the Private Members of the Class
Print	It prints all the details of the person (doctor or patients)

(*) You may add to those methods where you find necessary

<u>Class Appointment:</u>
This class handles the information and the operations for the Appointments. This class has a relationship with both the Doctor class and the Patient class.

Private variables:

- Time slot (by the hour). (*You can assume we're using the 24 hours system and that time goes from 00-> 24 from earliest to latest*). i.e: int hour. The clinic starts from 9:00 AM and closes at11:00 PM.
- Any relevant variable(s) you may wish to add

Public Methods:

METHOD NAME	DESCRIPTION
Add Appointment (Normal or Urgent)	Adding an appointment to the doctor's list for a given patient in a free available slot
Cancel Appointment	It cancels the patient's appointment under the request of the patient or if the doctor has an emergency
Modify Appointment	It modifies the patient's appointment based on his/her request.
Find Available Slot	It finds the available slot for a given patient for a given doctor.

(*) You may add to those methods where you find necessary

Class Patient : Inherits from Person.

Private variables:

- Last Visit: The last visit where the patient visited the clinic.
- E-Mail: The E-Mail address of the patient
- Phone Number: The phone number of the patient
- Any relevant variable(s) you may wish to add

Public Methods:

METHOD	DESCRIPTION
Setters and Getters	Setters and Getters of the Private Members of the Class
Display Patient Info	It displays this patient's info.

(*) You may add to those methods where you find necessary

Class Doctor: Inherits from Person.

Private variables:

- Counter: Number of appointments the Dr has for the day.
- 1D data structure (appointment): that indicates the time slots the Dr is booked at.

- Any necessary extra information about a doctor.

Public Methods:

METHOD NAME	DESCRIPTION
Setters and Getters	Setters and Getters of the Private Members of the Class
Is Doctor Available ?	A function to return if the doctor is available at a certain time. The availability can be also handled on the Appointment class; as an alternative.

(*) You may add to those methods where you find necessary

Deliverables:

- Your application should simulate a complete working day.
- Your application is preferable to be menu driven (i.e. the clinic admin will have a menu where s/he can select various clinic operations like making an appointment to a patient in a given time)
- Please make sure to document your code by adding comments that explain your logic and the functions you're creating whenever possible. Useful URLs:
    - https://users.ece.cmu.edu/~eno/coding/CppCodingStandard.html
    - https://isocpp.org/wiki/faq/coding-standards
    - http://web.mit.edu/6.s096/www/standards.html

Suggested Test Case:

Patients:

Ahmed at 1:00, Sara at 4:00, and Kareem at 3:00, Mohammed 1:00

Doctors :

Ayman, Khaled, Mai

- Ahmed should be assigned to Dr Ayman 9:00
- Sara should be assigned to Dr Khaled at 10:00
- Kareem should be assigned to Dr Mai at 16:00
- Mohammed should be assigned to Dr Khaled at 10:00 (since Dr Ayman is unavailable at 10:00 and Dr Khaled is next).
- Sami should be assigned to Dr. Mai at 17:00 while Kamal has an urgent appointment at 17:00 so he will be taken this appointment while Sami will be allocated any available slot at Dr. Mai.

(*) You should create your own additional test cases to demonstrate the clinic's operations like cancelling an appointment by a doctor or a patient

Bonus: Doctors' Schedule Optimization:

- This is an enhanced version of the allocation where the doctors' schedule is optimized such that the patients will not have the option to choose their times. The service is a 'walk-in' service, based on a first-come-first serve. Accordingly, when the patient visits the clinic, s/he will be allocated to the available doctor.
- Modify your appointments' allocation algorithm accordingly.
- For example, the following demonstration allocates 11 patients to the doctors in a sequential order to keep all the doctors' schedule busy "without" considering patients' preferences.