1. (20 marks) Sketch the UML diagrams corresponding to the following situations:

a) (8 marks) Each patient with cancer is assigned for treatment to one cancer clinic, and is under the care of one supervising physician and two or three consulting physicians. Each physician can be associated with up to three cancer clinics, and can be either a consulting or supervising physician for any number of patients.

**Answer:** In my original answer, in which I assumed that a physician could be both a consulting and a supervising physician relative to different patients, consulting and supervision were encoded as relationships to patients with a single Physician class, *i.e.*,

```
Clinic --0..3-> Physician
          |           |
         /           \
    Consulting     Supervising
          |           |
Patient  <=*<=*=> Patient
```

However, as I was marking the exam, I realized that from the “either / or” wording in the last sentence of the description, it was equally valid to conclude that physician had to be one or the other, which would require sub-types of physicians, *i.e.*,

```
Clinic --0..3-> Physician
           |              |
          /                \
   PhysicianCon     PhysicianSpr
          |
        *                *
Patient
```

Which answer is correct, then? Both are, for the purposes of this exam; however, only one would be correct in a real-world system. This highlights the
need to resolve such ambiguities during design by re-consulting the system’s eventual users as necessary.

b) (12 marks) Every medical school has an associated set of registered medical students. Each medical student has an associated name and ID #, and can be either a 1st year student, a 2nd year student, or an intern. As students progress through their studies in groups, each student is associated with a class, which in turn has an associated expected year of graduation for the members of that class, e.g., MUN Medical School Class of 2012. Each 2nd year student has an integer rank that they achieved in their 1st year studies, and each intern has both an integer rank that they achieved in their 2nd year studies and a supervising doctor. Each intern is supervised by one doctor, and a doctor can supervise between 2 and 7 interns.

Answer: My original answer looked like this:

```
MedStudent MedSchool
  Name
  ID

MedStudYr1 MedStudYr2 MedStudInt
  RankYr1
  RankYr2

MedStudYr1 MedStudYr2
  1

MedClass
  1

Doctor
  2..7

Supervise
```

However, several of you pointed out that as long as we have a general Medical Student class, it is much cleaner design-wise to associate graduating classes with those general medical-student objects (and, moreover, to have graduating year associated with each graduating class), i.e.,

```
MedClass MedStudent MedSchool
  GradYr
  Name
  ID

Doctor
  1

Supervise
  2..7

MedStudYr1 MedStudYr2 MedStudInt
  RankYr1
  RankYr2
```
2. (30 marks)

All parts of this questions refer to the clinic-example handout.

a) (16 marks) Given classes Schedule, Appointment, Treatment, TreatmentRad, TreatmentChm, ClinicPersonnel, ClinPerNurse, ClinPerSpec, ClinPerSpecOnc, and ClinPerSpecRad derived from the description in the handout, give a UML diagram showing all of the relationships between these classes implicit in this description. Do not include class fields in this diagram.

Answer:

```
Appointment * Schedule

Treatment

TreatmentChm  TreatmentRad

ClinPerNurse  ClinPerSpecOnc  ClinPerSpecRad

ClinPersonnel
```

b) (14 marks) Specify the syntax portion of a MIS for the schedule module described in the handout by filling in the table on the following page. You may assume the existence of classes ClinicPersonnel and Appointment and exception-types InvalidCIN, InvalidDate, InvalidDateRange, InvalidRoom, InvalidApptDate, InvalidApptRoom, and InvalidReference.
<table>
<thead>
<tr>
<th>Name</th>
<th>Inputs</th>
<th>Outputs</th>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td>–</td>
<td>Schedule</td>
<td>–</td>
</tr>
<tr>
<td>makeAppt</td>
<td>APPOINTMENT</td>
<td>–</td>
<td>INVALIDReference, INVALIDApptDate, INVALIDApptRoom</td>
</tr>
<tr>
<td>getApptTimes</td>
<td>String, String, String</td>
<td>String[][2]</td>
<td>INVALIDDate, INVALIDDateRange, INVALIDRoom</td>
</tr>
<tr>
<td>getPerAppts</td>
<td>CLINICPERSONNEL</td>
<td>APPOINTMENT[]</td>
<td>INVALIDReference</td>
</tr>
<tr>
<td>delApptTimes</td>
<td>String, String</td>
<td>–</td>
<td>INVALID Date, INVALIDDateRange</td>
</tr>
<tr>
<td>delApptRoom</td>
<td>String</td>
<td>–</td>
<td>INVALIDRoom</td>
</tr>
</tbody>
</table>
The table on the previous page gives the answer to the question as stated. However, several of you pointed out the following problems and ambiguities with this statement:

- When we try to insert a new appointment into a schedule in method `makeAppt`, if the new appointment is in a room that is already booked for some part of the suggested time, then we merely know that the appointment is invalid, and cannot decide whether it was the suggested room or time was invalid; and

- A clear distinction needs to be made between an exception issued when a given room-number or date string is formatted wrong and when the room-number and date are invalid in the context of an a suggested appointment relative to a particular schedule, and having exception names that are very similar (and given without the precise conditions under which they are issued) as was done here does not make this distinction very clear.

Given the exceptions as stated, my answer is correct but it is an admittedly bad design in spots; this highlights the need to be very careful when adding error-handling to a system design.