United International University
CSE 1116 - Mid Exam - Set B

Write your name and ID on the question paper. Marks are on the right. Return question.

Hospital Management System

A hospital wants to build a simple object-oriented system to manage its people and services. You are asked to design classes following the principles of OOP.

Classes and Requirements

1. **Person (Base Class) [3]**
    - **Private fields:** String name, int age.
    - Getters and setters for both private fields. For age, check if (0 < age < 120).
    - String getDetails() - returns a string in the format "Person: name, age".
    - A constructor that takes 2 parameters: name and age.

2. **Patient (Subclass of Person) [2]**
    - **Field:** String disease.
    - A constructor that takes 3 parameters: name, age, disease.
    - Override the getDetails() method to return "Patient: name, age, disease".

3. **Doctor (Subclass of Person) [6]**
    - **Fields:** String specialization, Patient [ ] assignedPatients (size = 5).
    - A constructor that takes 3 parameters: name, age, specialization.
    - Override the getDetails() method to return "Doctor: name, age, specialization".
    - **Method:** addPatient(Patient p) – add the Patient p object in the assignedPatients array.
    - **Method:** patientInfo() – print details of all assigned patients.

4. **Treatment [3]**
    - **Private fields:** String treatmentCode, String description.
    - A constructor which accepts (treatmentCode, description).
    - Another constructor which accepts only (treatmentCode) with "General Checkup" as default description.
    - String treatmentInfo() – returns "Treatment: treatmentCode - description".

5. **Appointment [4]**
    - **Fields:** Patient patient, Doctor doctor, Treatment treatment, double fee (default -1 for not paid).
    - A constructor which takes (Patient, Doctor, Treatment).
    - **Method:** void assignFee(double f) (only valid if $f \geq 0$, otherwise ignore).
    - **Method:** String getReport() – returns "patientName treated by doctorName for treatmentDescription: fee" (if fee not assigned, show "Pending").

Main Tasks [7]

Create a **Main** class and do the following tasks in the **main** method:

1. Create at least **4 Patients** and **2 Doctors**, store them in a Person[], and print their details polymorphically.
2. Assign **2 Patients** to **each Doctor** as assigned patients in any order.
3. Create **2 Treatments** using different constructors and print their info.
4. Create at least **2 Appointments** by associating **Patients, Doctors, and Treatments**.
5. **Assign fees** to some appointments (show valid and invalid cases).
6. Print all **appointment** reports.
7. **Print** the **information** of the patients assigned to **both Doctors**.