Qatar University, College of Engineering, Computer Science and Engineering Department

Course: CMPS251 – Object Oriented Programming **Instructor:** ______________________

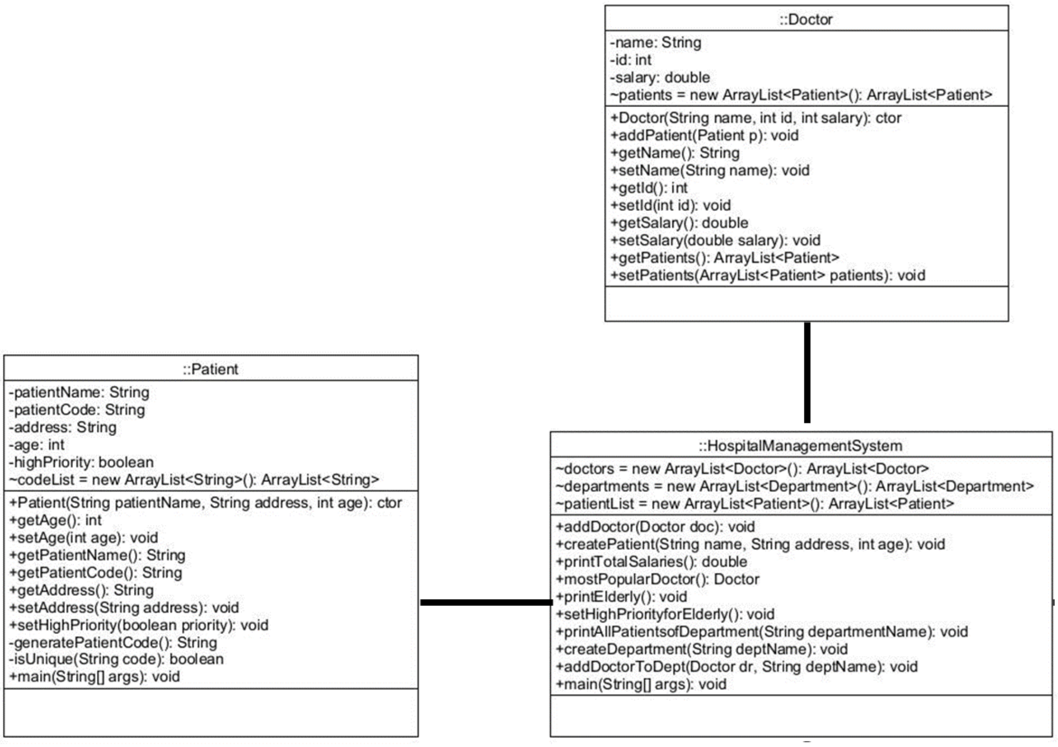
Midterm Exam

Term: Fall 2022, **Date:** Monday, November 7th, 2022, **Duration:** 60 minutes

Name: ____________________________________ **ID**: _________________________ **Score**: ………/25

Question 1: [15 pts (5 each), ~30 mins]

Consider the following hospital management system consists of 3 classes; `Doctor`, `Patient`, and `HospitalManagementSystem` as illustrated in the class diagram below:


::Doctor
-name: String
-id: int
-salary: double
~patients = new ArrayList<Patient>(): ArrayList<Patient>
+Doctor(String name, int id, int salary): ctor
+addPatient(Patient p): void
+getName(): String
+setName(String name): void
+getId(): int
+setId(int id): void
+getSalary(): double
+setSalary(double salary): void
+getPatients(): ArrayList<Patient>
+setPatients(ArrayList<Patient> patients): void
::Patient
-patientName: String
-patientCode: String
-address: String
-age: int
-highPriority: boolean
~codeList = new ArrayList<String>(): ArrayList<String>
+Patient(String patientName, String address, int age): ctor
+getAge(): int
+setAge(int age): void
+getPatientName(): String
+getPatientCode(): String
+getAddress(): String
+setAddress(String address): void
+setHighPriority(boolean priority): void
-generatePatientCode(): String
-isUnique(String code): boolean
+main(String[] args): void
::HospitalManagementSystem
~doctors = new ArrayList<Doctor>(): ArrayList<Doctor>
~departments = new ArrayList<Department>(): ArrayList<Department>
~patientList = new ArrayList<Patient>(): ArrayList<Patient>
+addDoctor(Doctor doc): void
+createPatient(String name, String address, int age): void
+printTotalSalaries(): double
+mostPopularDoctor(): Doctor
+printElderly(): void
+setHighPriorityforElderly(): void
+printAllPatientsofDepartment(String departmentName): void
+createDepartment(String deptName): void
+addDoctorToDept(Doctor dr, String deptName): void
+main(String[] args): void

Complete `TODO 1` in the class `Patient`, and `TODOs 2` and 3 in the class `HospitalmanagementSystem`.

- Assume that all getters and setters are appropriately coded for all private attributes of all classes.
- The static method `Math.Random()` returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
- You can add methods/attributes as you see fit.
- The output of running the `main` method is provided after the code.

Doctor.java

```java
import java.util.ArrayList;
public class Doctor {
	private String name;
	private int id;
	private double salary;
	// a list of patients the doctor currently assigned to.
	ArrayList<Patient> patients=new ArrayList<Patient>();

	public void addPatient(Patient p){
		patients.add(p);
	}

	public Doctor(String name, int id,int salary) {
		super();
		this.name = name;
		this.id = id;
		this.salary=salary;
	}
}
```

Patient.java

```
import java.util.ArrayList;

public class Patient {
      private String patientName;
      private String patientCode; // unique identification number consisting of 10 digits
      private String address;
      private int age;
      private boolean highPriority; // true if high priority, false if not
      static ArrayList<String> codeList=new ArrayList<String>();

      public Patient(String patientName, String address, int age) {
            this.patientName = patientName;
            this.patientCode = generatePatientCode();
            this.address = address;
            this.age=age;

            highPriority=false;
      }

      /*TODO 1: Create the method to generate the patientCode, which consists
        of 10 randomly generated numbers.
        The generated code should be unique for each patient. */

      private String generatePatientCode()
      {




}
```

HospitalManagementSystem.java

```
import java.util.ArrayList;

public class HospitalManagementSystem {
        ArrayList<Doctor> doctors=new ArrayList<Doctor>();
        ArrayList<Department> departments =new ArrayList<Department>();
        ArrayList<Patient> patientList=new ArrayList<Patient>();

        public void addDoctor(Doctor doc)
        {
                doctors.add(doc);
        }

        //TODO 2: write the code for the method to print the total salaries of all doctors.
        public double printTotalSalaries(){



        }

        /* TODO 3: Write a method to return a reference to the most popular doctor. The most
           popular doctor is the one who has most appointments. If two doctors have the same
           number of patients, you may return a reference to any of them. */



        public static void main(String[] args)
```

```
    {
        HospitalManagementSystem system=new HospitalManagementSystem();

        Patient p=new Patient("Mohammed","Qatar",77);
        Patient p1=new Patient("Yuki","Japan",40);
        Patient p2=new Patient("Nusret","Turkey",50);
        Patient p3=new Patient("Michael","USA",20);
        Patient p4=new Patient("Ali","Iran",66);
        Patient p5=new Patient("Rabeh","Algeria",60);
        Patient p6=new Patient("Mathews","Australia",50);
        Patient p7=new Patient("Rashid","Jordan",199);

        Doctor d1=new Doctor("Amal", 50,500);
        system.addDoctor(d1);
        // Assigning patients to d1
        d1.addPatient(p);
        d1.addPatient(p1);
        d1.addPatient(p2);

        Doctor d2=new Doctor("Omar", 50,500);
        system.addDoctor(d2);
        // Assigning patients to d2
        d2.addPatient(p4);
        d2.addPatient(p5);
        d2.addPatient(p6);
        d2.addPatient(p7);

        Doctor d3=new Doctor("Eman", 50,400);
        system.addDoctor(d3);

        // Assigning patients to d3

        d3.addPatient(p3);
        d3.addPatient(p4);
        d3.addPatient(p5);
        d3.addPatient(p6);
        d3.addPatient(p7);

        // printing total salaries of the doctors
        system.printTotalSalaries();

        // Printing the name of the most popular doctor.
        System.out.print("The Most popular Doctor is: ");
        Doctor mostpopular= system.mostPopularDoctor();
        System.out.println(mostpopular.getName());


    }
}
```

The output of `main` is as follows:

```
Total salaries are 1400.0
The Most popular Doctor is: Eman
```

Question 2: [10 pts (2.5 each), ~30 mins] Consider the code of the classes Actor, Movie, and MovieDataSet, then code the methods below as public methods in the class **MovieDataSet**. Assume that all getters and setters are appropriately coded for all private attributes of all classes.

```java
public class Actor {
        private String name;
        private String gender; //"Male" or "Female"

        public Actor() {    }
        public Actor(String name, String gender) {
                setName(name);
                setGender(gender);
        }
}

import java.util.ArrayList;
public class Movie {
        private String title;
        private String director;
        private int year;
        private double rating;
        private double expenses;
        private double earnings;
        private ArrayList<Actor> actors = new ArrayList<Actor>();

        public Movie() {    }
        public Movie(String title, String director, int year, double rating,
                        double expenses, double earnings, ArrayList<Actor> actors) {
                setTitle(title);
                setDirector(director);
                setYear(year);
                setRating(rating);
                setExpenses(expenses);
                setEarnings(earnings);
                setActors(actors);
        }
}

import java.util.ArrayList;
public class MoviesDataSet {
        private ArrayList<Movie> movies = new ArrayList<Movie>();
        public MoviesDataSet() {  }
        public MoviesDataSet(ArrayList<Movie> movies) { setMovies(movies); }
        public ArrayList<Movie> getMovies() { return movies; }
        public void setMovies(ArrayList<Movie> movies) { this.movies = movies; }
        // (a)
        // (b)
        // (c)
        // (d)
}
```

(a) The method moviesByDirector that takes a director's name DirectorName and returns a list of Movie objects having that the movie is directed by the director DirectorName.

```
public ArrayList<Movie> moviesByDirector(String DirectorName) {
      ArrayList<Movie> result = new ArrayList<Movie>();







      return result;
}
```

(b) The method moviesByActor that takes an actor's name ActorName and returns a list of Movie objects having that the actor ActorName is acting in the movie.

```
public ArrayList<Movie> moviesByActor(String ActorName) {
      ArrayList<Movie> result = new ArrayList<Movie>();







      return result;
}
```

(c) The method gainedProfitMovies, which returns a list of movies' titles where the earnings of the movie is more than the expenses of the movie.

```
public ArrayList<String> gainedProfitMovies() {
       ArrayList<String> result = new ArrayList<String>();






       return result;
}
```

(d) The method highestRatingMovie that takes the year year and returns the title of the movie having the highest rating in that year. If the value of year passed to this method is zero, then the method returns the title of the movie having the highest rating regardless of the year.

```
public String highestRatingMovie(int year) {
       String result = "";






       return result;
}
```