

## Tutorial 9(b) Polymorphisms

### Problem

Consider the class diagram in Figure 1 which illustrates the data model for patient records in a hospital. There are two types of patients; inpatients which need to be admitted to wards (thus the information about their wards will be recorded) and outpatients which do not need to stay at the hospital. As for each outpatient, the information regarding the type of diagnosis needed will also be recorded. Regardless of either inpatients or outpatients, the hospital needs to record the patient's illness and the information about his or her guardian.

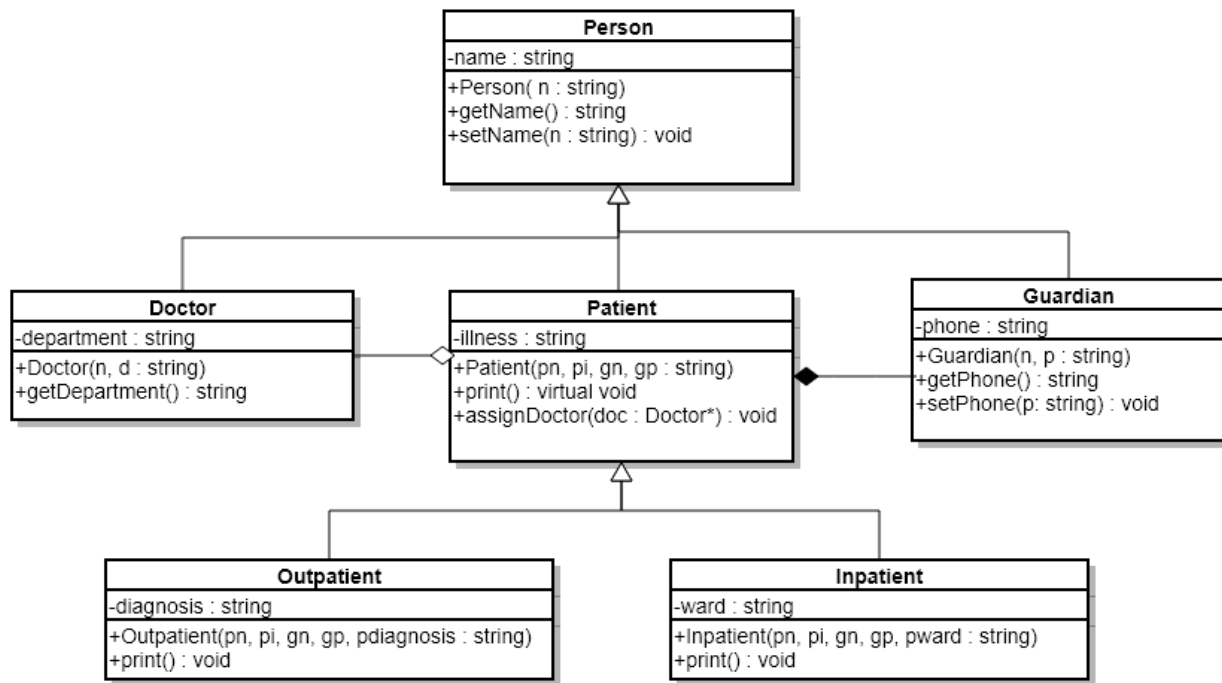


Figure 1: Class diagram for patient records.

Based on the class diagram in Figure 1, complete the given program (**tutorial\_9b.cpp**) to perform the following tasks:

- Implement all the classes (except the class **Person**) with the member data or attributes and member functions or methods specified in the diagram. Note that, the definition for the class **Person** is fully given, whereas the definitions for the other classes are partially given. The purpose of each method is as the name implies and some of them are further explained below:
  - The method **assignDoctor** in the class **Patient** will be used to assign a doctor to a patient.

- The method **print** in the class **Patient**, **Outpatient**, and **Inpatient** will be used to print the patient's information onto the screen, accordingly.
- Your implementation must apply some OOP principles including **aggregations**, **compositions**, **inheritances** and **polymorphisms**.
  - Create an array to hold a list of doctors and initialize the array with the following data:

Doctor's Name	Department
Dr. Ramli	ICU
Dr. Kamariah	Radiology

- Create an array to hold a list of patients below. You **must use only a single array** for this.

Patient's Name	Illness	Guardian's Name	Guardian's Phone	Diagnosis Needed	Ward Number	Patient Type
Rozita	Sprained ankle	Salleh	4466	X-ray	-	Outpatient
Nurdiana	Respiratory failure	Jalil	7731	-	ICU 101	Inpatient
Ali	Coronary artery	Bakar	1234	CT Scan	-	Outpatient

- Your program needs to provide the user a menu-driven interaction with the following options.

Menu Options	Description
1. List Doctors	To print all doctors onto the screen.
2. List Patients	To print all patients onto the screen.
3. Assign Doctor	To assign a patient with a doctor. The user needs to choose the respective patient and doctor using their array indices. As such, the program needs to ensure that the indices are within the valid range. Otherwise, an exception indicating out of range will be thrown. You <b>must use the exception handling</b> strategy for this purpose.
4. Exit	To end the program.

Figure 2 shows the expected result of the program. Note that, all the interactions shown in the figure are continuous in a single run. Note also that, the **bold** texts indicate input entered by the user. Meanwhile, Table 1 shows the assessment criteria for this question.

*Interaction 1: The user chooses option 1 to list the doctors.*

```

===== Menu =====
1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

```

Choose an operation [1-4] => **1**

Doctor's Name: Dr. Ramli	Department: ICU
Doctor's Name: Dr. Kamariah	Department: Radiology

*Interaction 2: The user chooses option 2 to list the patients.*

===== Menu =====

1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

Choose an operation [1-4] => **2**

Patient #1

Patient Type: OUTPATIENT

Diagnosis : X-ray

Patient's Name :Rozita

Illness :Sprained ankle

Guardian's Name :Salleh

Guardian's Phone :4466

\*\*\* No doctor assigned yet \*\*\*

Patient #2

Patient Type: INPATIENT

Ward : ICU 101

Patient's Name :Nurdiana

Illness :Respiratory failure

Guardian's Name :Jalil

Guardian's Phone :7731

\*\*\* No doctor assigned yet \*\*\*

Patient #3

Patient Type: OUTPATIENT

Diagnosis : CT Scan

Patient's Name :Ali

Illness :Coronary artery

Guardian's Name :Bakar

Guardian's Phone :1234

\*\*\* No doctor assigned yet \*\*\*

*Interaction 3: The user chooses option 3 to assign doctors to patients.*

===== Menu =====

1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

Choose an operation [1-4] => **3**

```
Enter the patient index and doctor index => 1 0
```

```
===== Menu =====
```

1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

```
Choose an operation [1-4] => 3
```

```
Enter the patient index and doctor index => 2 1
```

*Interaction 4: The user chooses option 3 to assign a doctor but he or she enters **incorrect array index** for the patient or the doctor.*

```
===== Menu =====
```

1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

```
Choose an operation [1-4] => 3
```

```
Enter the patient index and doctor index => 5 0
```

```
** Error: Index is out of range
```

```
===== Menu =====
```

1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

```
Choose an operation [1-4] => 3
```

```
Enter the patient index and doctor index => 0 -1
```

```
** Error: Index is out of range
```

*Interaction 5: The user chooses option 5 to list again the patients*

```
===== Menu =====
```

1. List Doctors
2. List Patients
3. Assign Doctor
4. Exit

```
Choose an operation [1-4] => 2
```

```
Patient #1  
Patient Type:  OUTPATIENT  
Diagnosis    :  X-ray
```

```
Patient's Name      :Rozita
Illness             :Sprained ankle
Guardian's Name     :Salleh
Guardian's Phone    :4466
*** No doctor assigned yet ***

Patient #2
Patient Type:      INPATIENT
Ward               :   ICU 101

Patient's Name      :Nurdiana
Illness             :Respiratory failure
Guardian's Name     :Jalil
Guardian's Phone    :7731
Doctor's Name       :Dr. Ramli
Doctor's Dept.      :ICU

Patient #3
Patient Type:      OUTPATIENT
Diagnosis          :   CT Scan

Patient's Name      :Ali
Illness             :Coronary artery
Guardian's Name     :Bakar
Guardian's Phone    :1234
Doctor's Name       :Dr. Kamariah
Doctor's Dept.      :Radiology
```

**Figure 2:** An example run of the program.