

CSC 113

Tutorial 4

Relationship Between Classes 2

Patient

-id : int

-name : String

-age : int

+Patient(id: int , name: String, age: int)

+Patient(p : Patient)

+Setters/Getters

+display() : void

Constructor should assign values to corresponding attributes.

```
public class Patient {  
    private int id;  
    private String name;  
    private int age;
```

```
    public Patient(int id, String name, int age)  
    {  
        this.id = id;  
        this.name = name;  
        this.age = age;  
    }  
}
```

Patient
-id : int
-name : String
-age : int
+Patient(id: int , name: String, age: int)
+Patient(p : Patient)
+Setters/Getters
+display() : void

Second constructor should copy values to corresponding attributes.

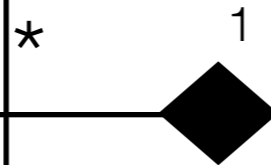
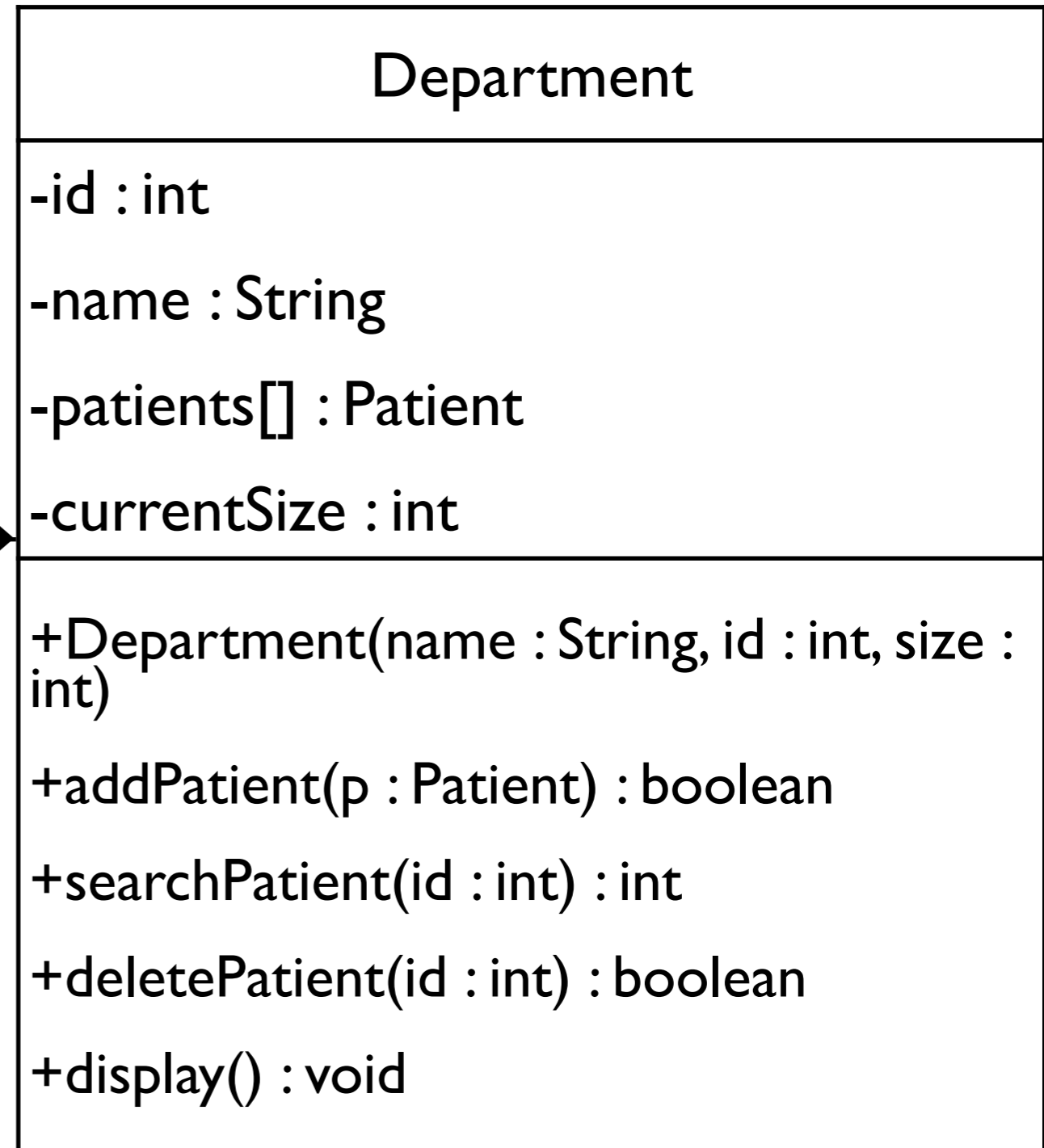
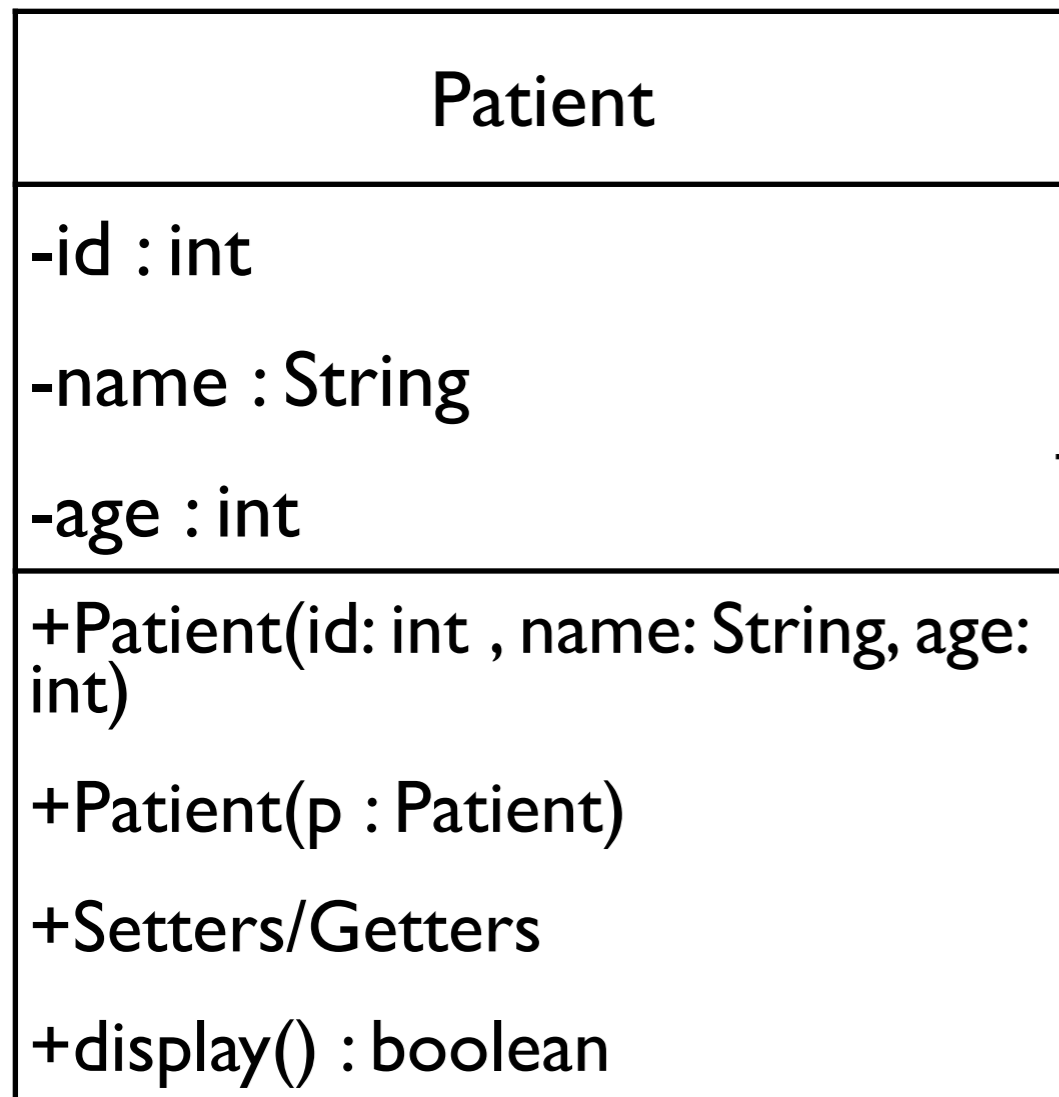
```
public Patient(Patient p)
{
    id = p.getId();
    name = p.getName();
    age = p.getAge();
}
```

Patient
-id : int
-name : String
-age : int
+Patient(id: int , name: String, age: int)
+Patient(p : Patient)
+Setters/Getters
+display() : void

display should print the values.

```
public void display()  
{  
    System.out.println(id+" \t"+name+" \t"+age);  
}
```

Patient
-id : int
-name : String
-age : int
+Patient(id: int , name: String, age: int)
+Patient(p : Patient)
+Setters/Getters
+display() : void



Constructor should assign values to corresponding attributes.

```
public Department(String name, int id, int
size)
{
    this.name = name;
    this.id = id;
    patients = new Patient[size];
}
```

Department
-id : int -name : String -patients[] : Patient -currentSize : int
+Department(name : String, id : int, size : int) +addPatient(p : Patient) : boolean +searchPatient(id : int) : int +deletePatient(id : int) : boolean +display() : void

AddPatient takes a patient and copy it to the department.

```
public boolean addPatient(Patient s)
{
    if(currentSize < patients.length)
    {
        patients[currentSize++] = new Patient(s);
        return true;
    }
    return false;
}
```

Department
-id : int
-name : String
-patients[] : Patient
-currentSize : int
+Department(name : String, id : int, size : int)
+addPatient(p : Patient) : boolean
+searchPatient(id : int) : int
+deletePatient(id : int) : boolean
+display() : void

Search for the given id. return the index if found. otherwise return -1.

```
public int searchPatient(int patId)
{
    int i;
    for(i=0; i<currentSize; i++)
        if(patients[i].getId() == patId)
            return i;
    return -1;
}
```

Department
-id : int
-name : String
-patients[] : Patient
-currentSize : int
+Department(name : String, id : int, size : int)
+addPatient(p : Patient) : boolean
+searchPatient(id : int) : int
+deletePatient(id : int) : boolean
+display() : void

Searches for the given id and deletes it and returns true. If not found returns false.

```
public boolean deletePatient(int id)
{
    int i = searchPatient(id);

    if(i == -1)
        return false;

    patients[i] = patients[--currentSize];
    patients[currentSize] = null;
    return true;
}
```

Department
-id : int
-name : String
-patients[] : Patient
-currentSize : int
+Department(name : String, id : int, size : int)
+addPatient(p : Patient) : boolean
+searchPatient(id : int) : int
+deletePatient(id : int) : boolean
+display() : void

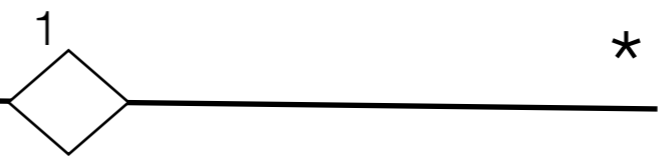
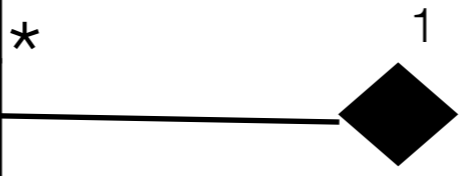
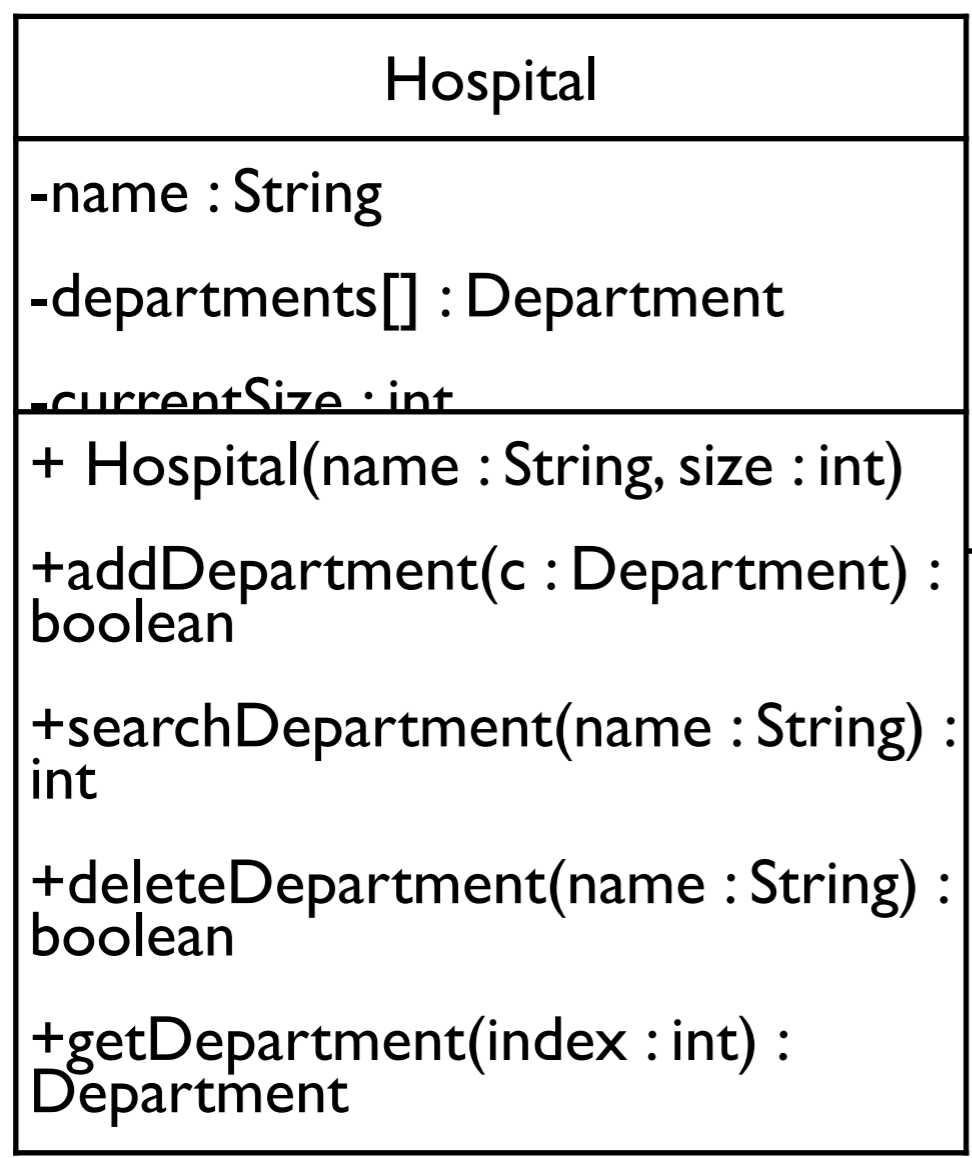
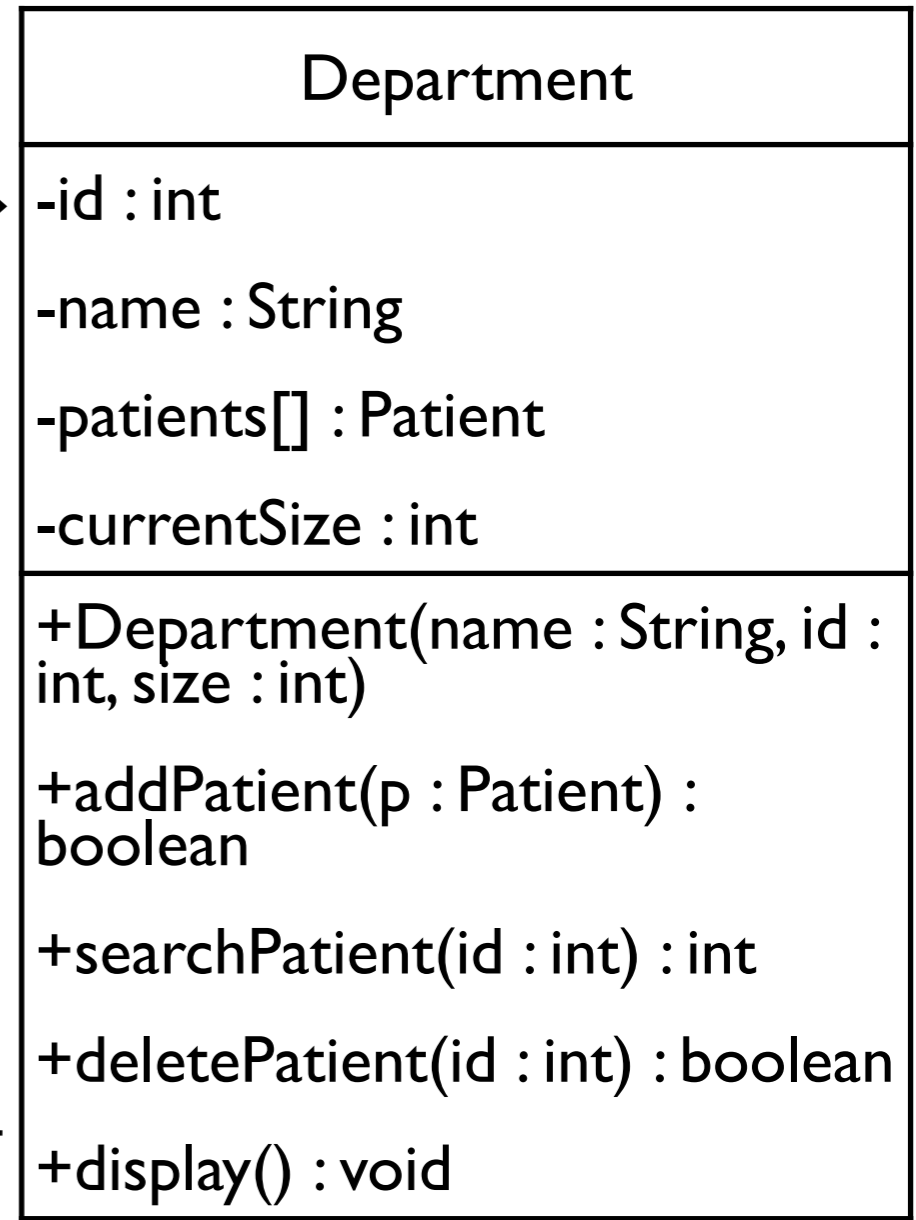
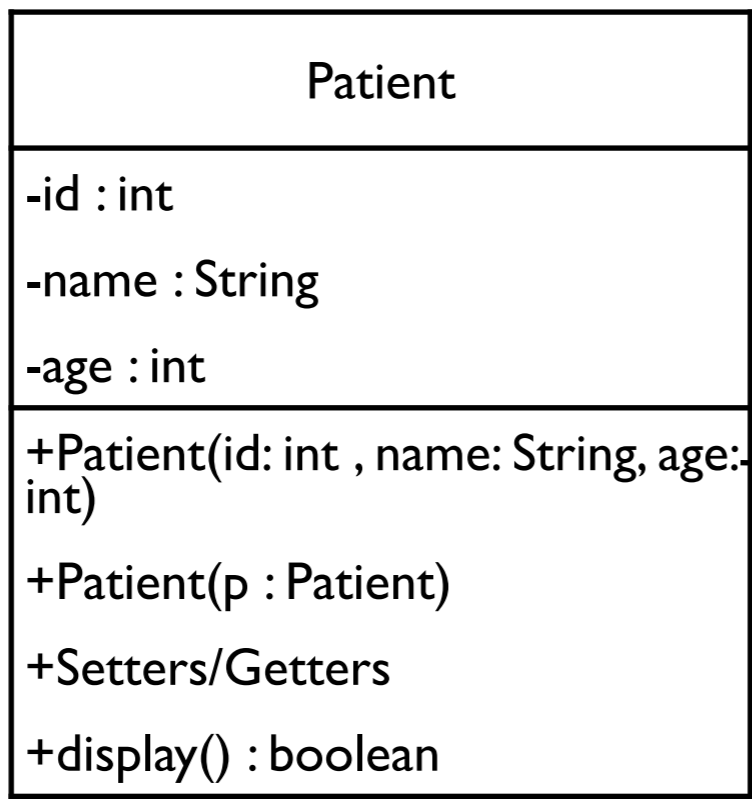
display the name and id of the department and all patients in it

```
public void displayAll()
{
    int i;

    System.out.println("Department ID: " + id + "\t\tDepartment Name:
"+ name);
    System.out.println("ID \t\t Name \t\t\t Age");

    for(i=0; i<currentSize; i++)
        patients[i].display();
}
```

Department
-id : int
-name : String
-patients[] : Patient
-currentSize : int
+Department(name : String, id : int, size : int)
+addPatient(p : Patient) : boolean
+searchPatient(id : int) : int
+deletePatient(id : int) : boolean
+display() : void



Constructor should assign values to corresponding attributes.

```
public Hospital(String name, int size)
{
    this.name = name;
    departments = new Department[size];
}
```

Hospital
-name : String -departments[] : Department -currentSize : int
+ Hospital(name : String, size : int) +addDepartment(d : Department) : boolean +searchDepartment(name : String) : int +deleteDepartment(name : String) : boolean +getDepartment(index : int) : Department +displayAll() : void

AddDepartment takes a department and add it to the hospital.

```
public boolean addDepartment(Department d)
{
    if(currentSize < departments.length)
    {
        departments[currentSize++]=d;
        return true;
    }
    return false;
}
```

Hospital
-name : String -departments[] : Department -currentSize : int
+ Hospital(name : String, size : int) +addDepartment(d : Department) : boolean +searchDepartment(name : String) : int +deleteDepartment(name : String) : boolean +getDepartment(index : int) : Department +displayAll() : void

Search for the given name. returns the index if found. otherwise returns -1.

```
public int searchDepartment(
    String dName)
{
    int i;

    for(i=0; i<currentSize; i++)
        if(departments[i].getName().equalsIgnoreCase(dName))
            return i;
    return -1;
}
```

Hospital
-name : String -departments[] : Department -currentSize : int
+ Hospital(name : String, size : int) +addDepartment(d : Department) : boolean +searchDepartment(name : String) : int +deleteDepartment(name : String) : boolean +getDepartment(index : int) : Department +displayAll() : void

Searches for the given name and deletes it and returns true. If not found returns false.

```
public boolean deleteDepartment(
    String dName)
{
    int i =searchDepartment(dName);

    if(i==-1)
        return false;

    departments[i] = departments[--currentSize];
    departments[currentSize] = null;
    return true;
}
```

Hospital
-name : String
-departments[] : Department
-currentSize : int
+ Hospital(name : String, size : int)
+addDepartment(d : Department) : boolean
+searchDepartment(name : String) : int
+deleteDepartment(name : String) : boolean
+getDepartment(index : int) : Department
+displayAll() : void

display the name of the hospital and all departments and patients in it

```
public void displayAll()
{
    int i;
    for(i=0; i<currentSize; i++)
        departments[i].displayAll();
}
```

Hospital
-name : String -departments[] : Department -currentSize : int
+ Hospital(name : String, size : int) +addDepartment(d : Department) : boolean +searchDepartment(name : String) : int +deleteDepartment(name : String) : boolean +getDepartment(index : int) : Department +displayAll() : void

getDepartment return the given department index in the array.

```
public Department getDepartment(int i)
{
    if(i >= 0)
        return departments[i];
    return null;
}
```

Hospital
-name : String -departments[] : Department -currentSize : int
+ Hospital(name : String, size : int) +addDepartment(d : Department) : boolean +searchDepartment(name : String) : int +deleteDepartment(name : String) : boolean +getDepartment(index : int) : Department +displayAll() : void

Main

- we will use the following input

ICU
ID: 0

ENT
ID: 1

A&E
ID: 2

id	name	age
1	Ali	20
2	Majed	47
3	Saleh	18

id	name	age
4	Fahad	25
5	Bandr	15
6	Fawaz	32

id	name	age
7	Naif	16
8	Ahmed	51
9	Saud	63

Menu

- should look like this:

1- Add department.

2- Select department.

3- Delete department.

4- Display all.

5- Exit.

Select Department:

1- ICU

2- ENT

3- A&E



1- Add patient

2- Delete patient

3- Edit patient

4- Display patients

5- back to main menu

Creating the initial data

- Create all patients.
- Create departments and add the patients.
- Create the Hospital and add the departments.
- Now start creating the menu.

Creating The Menu

- Start with declaring variables for input.
- Start the loop. (do{}while();)
- Display the menu.
- Add switch statement for each item in the menu.
- Always have a default case for wrong input.
- To create a sub menu, use the same steps in one of the cases.

Select operation

- The method `getDepartment` will help but it is not the only way.
- Use it to print the names of the departments and to have access to `Department` class operations.