

## LAB # 1

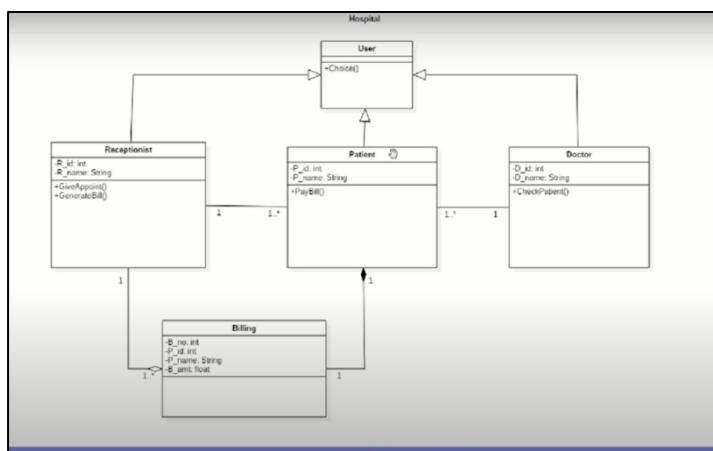
### Introduction to Eclipse IDE

#### OBJECTIVE

Introduction to Eclipse IDE, creating projects, its classes and mapping UML class diagram in to java code.

#### Lab Task:

Write the java code with the help of the given class diagram of Hospital Management System.



#### **CODE:**

```
Task1.java ×
1 package SCDLab1; // Ensure you're using the package if needed.
2
3 public class Task1 {
4
5     // Define the User class
6     static class User {
7         void choice() {
8             System.out.println("Choose your Role: Doctor, Patient, or Receptionist");
9         }
10    }
11
12    // Define the Patient class
13    static class Patient extends User {
14        int p_id;
15        String p_name;
16
17        Patient(int id, String name) {
18            this.p_id = id;
19            this.p_name = name;
20        }
21
22        void payBill() {
23            System.out.println(p_name + " is paying the bill.");
24        }
25    }
26
27    // Define the Doctor class
28    static class Doctor extends User {
29        int d_id;
30        String d_name;
31
32        Doctor(int id, String name) {
33            this.d_id = id;
34            this.d_name = name;
35        }
36    }
37
38    // Define the Billing class
39    static class Billing extends User {
40        int B_no;
41        int B_id;
42        String B_name;
43        float B_amt;
44
45        Billing(int no, int id, String name, float amt) {
46            this.B_no = no;
47            this.B_id = id;
48            this.B_name = name;
49            this.B_amt = amt;
50        }
51
52        void printBill() {
53            System.out.println("Bill Details: " + B_name + ", Amount: " + B_amt);
54        }
55    }
56
57    public static void main(String[] args) {
58        User user = new Patient(1, "John Doe");
59        user.choice();
60        user.payBill();
61    }
62}
```

## Lab # 1

```
Task1.java X
35     }
36
37     void checkPatients() {
38         System.out.println(d_name + " is checking patients.");
39     }
40 }
41
42 // Define the Receptionist class
43 static class Receptionist extends User {
44     int r_id;
45     String r_name;
46
47     Receptionist(int id, String name) {
48         this.r_id = id;
49         this.r_name = name;
50     }
51
52     void giveAppointment() {
53         System.out.println(r_name + " is giving appointments.");
54     }
55
56     void generateBill() {
57         System.out.println("The bill is generating...");
58     }
59 }
60
61 // Define the Billing class
62 static class Billing {
63     int B_no;
64     int p_id;
65     String p_name;
66     float B_amt;
67
68     Billing(int no, int id, String name, float amt) {
69         this.B_no = no;
70         this.p_id = id;
71         this.p_name = name;
72         this.B_amt = amt;
73     }
74 }
75
76 // Define the Surgeon class, which extends Doctor
77 static class Surgeon extends Doctor {
78     Surgeon(int id, String name) {
79         super(id, name); // Calling the parent constructor (Doctor)
80     }
81
82     void surgeonDetails() {
83         System.out.println("Surgeon Details...");
84     }
85 }
86
87 // Main method for execution
88 public static void main(String[] args) {
89     // User choice is shown, though we don't use the base class directly
90     User u = new User();
91     u.choice();
92
93     // Create a patient and perform action
94     Patient p1 = new Patient(111, "Khurram Raza");
95     p1.payBill();
96
97     // Create a surgeon and perform actions
98     Surgeon s = new Surgeon(101, "Dr. Rehan Khan");
99     s.checkPatients();
100    s.surgeonDetails();
101 }
102 }
```

### OUTPUT:

```
Problems @ Javadoc Declaration Console X Coverage
<terminated> Task1 (1) [Java Application] C:\Program Files\Java\jdk-22\bin\javav
Choose your Role: Doctor, Patient, or Receptionist
Khurram Raza is paying the bill.
Dr. Rehan Khan is checking patients.
Surgeon Details...
```