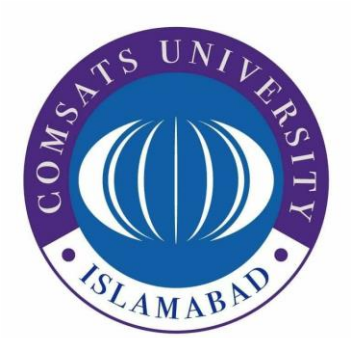
COMSATS University Islamabad
Attock Campus



Department Of Computer Science

Course	OOP
Instructor	Mr. Umar Zia
Lab Assignment No.	04
Title	Hospital Management system

Submitted by

Registration No.	**Name**
FA22-BAI-034	Mohsin Zia
FA22-BAI-049	Huzaifa
FA22-BAI-052	M. Anas Murtaza
FA22-BAI-055	Muhammad Amaan

HOSPITAL

```
import java.util.ArrayList;

public class Hospital {
    String hospital_name;
    ArrayList<Wards> wards;

    Hospital(String hospital_name,ArrayList<Wards> wards){
        this.hospital_name = hospital_name;
        this.wards = wards;
    }

}
```

PERSON

```
public class Person {
    private String name;
    private int age;
    private String contact;

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void setContact(String contact) {
        this.contact = contact;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getContact() {
        return contact;
    }

    public Person(String name, int age, String contact) {
        setName(name);
        setAge(age);
        setContact(contact);
    }
}
```

PATIENT

```
public abstract class Patient  extends Person {
    private String illness;
    private double fee;

    abstract double discounted_fee();

    public void setIllness(String illness) {
        this.illness = illness;
    }

    public String getIllness() {
        return illness;
    }

    public Patient(String name, int age, String contact, String illness,
double fee) {
        super(name, age, contact);
        setIllness(illness);
        setFee(fee);
    }

    public double getFee() {
        return fee;
    }

    public void setFee(double fee) {
        this.fee = fee;
    }

}
```

ADMITTED PATIENT

```
public class AdmittedPatient extends Patient {

    @Override
    double discounted_fee() {
        double discfee = getFee() - 500;
        return discfee;
    }

    public AdmittedPatient(String name, int age, String contact, String
illness, double fee) {
        super(name, age, contact, illness, fee);
    }

    void AdmittedPatient() {
        System.out.println("Name of Patient: " + getName());
        System.out.println("Age of Patient: " + getAge());
        System.out.println("Illness of Patient: " + getIllness());
        System.out.println("Patients Fee: " + discounted_fee());
    }
}
```

CHECKUP

```
public class Checkup extends Patient {
    public Checkup(String name, int age, String contact, String illness,
double fee) {
        super(name, age, contact, illness, fee);
    }

    char a;

    @Override
    double discounted_fee() {
        if (a == 'S') {
            double discfee = getFee() - 500;
            return discfee;
        } else {
            return getFee();
        }
    }

    public void Checkup() {
        System.out.println("Name of Patient: " + getName());
        System.out.println("Age of Patient: " + getAge());
        System.out.println("Illness of Patient: " + getIllness());
        System.out.println("Patients Fee: " + discounted_fee());
    }
}
```

EMPLOYEES

```
public abstract class Employees extends Person {
    private int emp_id;
    String designation;

    abstract double salary();

    public Employees(String name, int age, String contact, int emp_id, String
designation) {
        super(name, age, contact);
        setEmp_id(emp_id);
        setDesignation(designation);
    }

    public int getEmp_id() {
        return emp_id;
    }

    public String getDesignation() {
        return designation;
    }
```

```
    public void setDesignation(String designation) {
        this.designation = designation;
    }

    public void setEmp_id(int emp_id) {
        this.emp_id = emp_id;
    }
}
```

PARMANENT

```
public class Parmanent extends Employees implements Tax {
    double salary;

    public Parmanent(String name, int age, String contact, int emp_id, double
salary, String designation) {
        super(name, age, contact, emp_id, designation);
        setSalary(salary);
    }

    void Parmanent() {
        System.out.println("Employee name: " + getName());
        System.out.println("Age: " + getAge());
        System.out.println("Employee id: " + getEmp_id());
        System.out.println("Contact info: " + getContact());
        System.out.println("Designation: " + getDesignation());
        System.out.println("Yearly Salary: " + salary());
    }

    @Override
    double salary() {
        double yearlySalary = getSalary() * 12 - tax();
        return yearlySalary;
    }

    @Override
    public double tax() {
        double tax = ((3 * 100) / getSalary()) * 12;
        return tax;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public double getSalary() {
        return salary;
    }
}
```

CONTRACT

```
public class Contract extends Employees{
    private int hourlyRate;
    private int workingHoursPerDay;
    private int workingDaysPerMonth;

    public Contract(String name, int age, String contact, int emp_id, String
designation,int hourlyRate,int workingHoursPerDay, int workingDaysPerMonth) {
        super(name, age, contact, emp_id, designation);
        setHourlyRate(hourlyRate);
        setWorkingDaysPerMonth(workingDaysPerMonth);
        setWorkingHoursPerDay(workingHoursPerDay);
    }
    void Contract(){
        System.out.println("Employee name: " + getName());
        System.out.println("Age: " + getAge());
        System.out.println("Employee id: " + getEmp_id());
        System.out.println("Contact info: " + getContact());
        System.out.println("Designation: " + getDesignation());
        System.out.println("Enter: Hourly Rate, Working Hours per Day &
Working Days par Month:
"+getHourlyRate()+getWorkingHoursPerDay()+getWorkingDaysPerMonth());
        System.out.println("Yearly Salary: " + salary());
    }

    public int getWorkingHoursPerDay() {
        return workingHoursPerDay;
    }

    public void setWorkingHoursPerDay(int workingHoursPerDay) {
        this.workingHoursPerDay = workingHoursPerDay;
    }

    public double getHourlyRate() {
        return hourlyRate;
    }

    public int getWorkingDaysPerMonth() {
        return workingDaysPerMonth;
    }

    public void setWorkingDaysPerMonth(int workingDaysPerMonth) {
        this.workingDaysPerMonth = workingDaysPerMonth;
    }

    public void setHourlyRate(int hourlyRate) {
        this.hourlyRate = hourlyRate;
    }

    @Override
    double salary() {
        double monthlySalary = hourlyRate * workingHoursPerDay *
workingDaysPerMonth;
        double yearlySalary = monthlySalary * 12; // Assuming 12 months in a
year
```

```
        return yearlySalary;
    }
}
```

WARDS

```
import java.util.ArrayList;
public class Wards implements LoginWards {
    private String name;
    private int id;
    private int capacity;
    private String username;
    private int pin;
    ArrayList<AdmittedPatient> admittedPatients;

    Wards(String name, int id, int capacity, String username, int pin,
ArrayList<AdmittedPatient> patients) {
        setname(name);
        setPin(pin);
        setid(id);
        setUsername(username);
        setcapacity(capacity);
        admittedPatients = patients;
    }

    @Override
    public Boolean login(String username, int pin) {
        if (username.equals("doctor") && pin == 1234) {
            return Boolean.TRUE;
        } else {
            return Boolean.FALSE;
        }
    }

    public ArrayList<AdmittedPatient> getAdmittedPatients() {
        return admittedPatients;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public void setPin(int pin) {
        this.pin = pin;
    }

    public String getUsername() {
        return username;
    }

    public int getPin() {
        return pin;
    }
```

```
    public void setname(String name) {
        this.name = name;
    }

    public void setid(int id) {
        this.id = id;
    }

    public void setcapacity(int capacity) {
        this.capacity = capacity;
    }

    public String getname() {
        return this.name;
    }

    public int getid() {
        return id;
    }

    public int getcapacity() {
        return capacity;
    }

    @Override
    public String toString() {
        return "Wards{" +
                "name='" + name + '\'' +
                ", id=" + id +
                ", capacity=" + capacity +
                ", username='" + username + '\'' +
                ", password='" + pin + '\'' +
                ", admittedPatients=" + admittedPatients +
                '}';
    }
}
```

LOGINWARDS INTERFACE

```
public interface LoginWards {
    Boolean login(String username, int pin);
}
```

TAX

```
public interface Tax {
    double tax();
}
```

EMPLOYEEGUI

```
import java.util.Scanner;
        import java.awt.event.ActionEvent;
        import java.awt.event.ActionListener;
        import javax.swing.JButton;
        import javax.swing.JFrame;
        import javax.swing.JLabel;
        import javax.swing.JTextField;

public class EmployeeGUI implements ActionListener {
    JFrame f;
    JLabel l1, l2, l3, l4;
    JTextField tf1, tf2, tf3, tf4;
    JButton a, b;

    EmployeeGUI() {
        f = new JFrame("Employee");
        f.setSize(500, 500);
        f.setLocation(500, 0);

        l1 = new JLabel();
        l1.setText("Employee Name");
        l1.setBounds(50, 40, 100, 30);

        tf1 = new JTextField();
        tf1.setBounds(190, 40, 150, 30);

        l2 = new JLabel();
        l2.setText("Employee Age");
        l2.setBounds(50, 80, 100, 30);

        tf2 = new JTextField();
        tf2.setBounds(190, 80, 150, 30);

        l3 = new JLabel();
        l3.setText("Employee Designation");
        l3.setBounds(50, 120, 120, 30);

        tf3 = new JTextField();
        tf3.setBounds(190, 120, 150, 30);

        l4 = new JLabel();
        l4.setText("Salary");
        l4.setBounds(50, 160, 150, 30);

        tf4 = new JTextField();
        tf4.setBounds(190, 160, 150, 30);

        b = new JButton("Permanent");
        b.setBounds(140, 410, 100, 30);
        b.addActionListener(this);

        a = new JButton("Contract");
        a.setBounds(260, 410, 100, 30);
        a.addActionListener(this);
```

```
        f.add(a);
        f.add(b);
        f.add(l2);
        f.add(l1);
        f.add(tf1);
        f.add(tf2);
        f.add(l3);
        f.add(tf3);
        f.add(l4);
        f.add(tf4);

        f.setLayout(null);
        f.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Scanner scan = new Scanner(System.in);
        if (e.getSource() == a) {
            int sal = 50000;
            String result = String.valueOf(sal);
            tf4.setText(result);
        } else if (e.getSource() == b) {
            int sal = 80000;
            String result = String.valueOf(sal);
            tf4.setText(result);
        }
    }
}
```

PATIENTGUI

```
import java.awt.event.ActionEvent;
        import java.awt.event.ActionListener;
        import javax.swing.JButton;
        import javax.swing.JFrame;
        import javax.swing.JLabel;
        import javax.swing.JTextField;

public class PatientGUI implements ActionListener {
    JFrame f;
    JLabel l1, l2, l3, l4;
    JTextField tf1, tf2, tf3, tf4;
    JButton a, b;

    PatientGUI() {
        f = new JFrame("Patient");
        f.setSize(500, 500);

        l1 = new JLabel();
        l1.setText("Patient Name");
```

```
        l1.setBounds(50, 40, 100, 30);

        tf1 = new JTextField();
        tf1.setBounds(170, 40, 150, 30);

        l2 = new JLabel();
        l2.setText("Patient Age");
        l2.setBounds(50, 80, 100, 30);

        tf2 = new JTextField();
        tf2.setBounds(170, 80, 150, 30);

        l3 = new JLabel();
        l3.setText("Patient Illness");
        l3.setBounds(50, 120, 100, 30);

        tf3 = new JTextField();
        tf3.setBounds(170, 120, 150, 30);

        l4 = new JLabel();
        l4.setText("Result");
        l4.setBounds(50, 160, 150, 30);

        tf4 = new JTextField();
        tf4.setBounds(170, 160, 250, 30);

        b = new JButton("Normal");
        b.setBounds(150, 410, 80, 30);
        b.addActionListener(this);

        a = new JButton("Serious");
        a.setBounds(250, 410, 80, 30);
        a.addActionListener(this);
        f.add(a);
        f.add(b);
        f.add(l2);
        f.add(l1);
        f.add(tf1);
        f.add(tf2);
        f.add(l3);
        f.add(tf3);
        f.add(l4);
        f.add(tf4);

        f.setLayout(null);
        f.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == a) {
            String c = "Patient Must Be Admitted To Ward";
            String result = c;
            tf4.setText(result);
        } else if (e.getSource() == b) {
            String c = "Take Medicine From Pharmacy";
            String result = c;
```

```
            tf4.setText(result);
        }
    }
}
```