Question 1 : PayrollSystem.java

Code :

```
import java.util.Scanner;

class Employee {
    private int employeeId;
    private String employeeName;
    private String designation;

    public void setEmployeeId(int employeeId) {
        this.employeeId = employeeId;
    }
    public void setEmployeeName(String employeeName) {
        this.employeeName = employeeName;
    }
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    public int getEmployeeId() {
        return employeeId;
    }
    public String getEmployeeName() {
        return employeeName;
    }
    public String getDesignation() {
        return designation;
    }
    public double calculateBonus() { // Default implementation. It can be
overridden in derived classes
        return 0.0;
    }
    public double calculateWeeklySalary() {// Default implementation. It can
be overridden in derived classes
        return 0.0;
    }
    public void displayDetails() {
        System.out.println("Employee ID: " + employeeId);
        System.out.println("Employee Name: " + employeeName);
        System.out.println("Designation: " + designation);
        System.out.println("Weekly Salary: " + calculateWeeklySalary());
        System.out.println("Bonus: " + calculateBonus());
    }
}

class HourlyEmployee extends Employee {
    private double hourlyRate;
    private int hoursWorked;
```

```
    public void setHourlyRate(double hourlyRate) {
        this.hourlyRate = hourlyRate;
    }
    public void setHoursWorked(int hoursWorked) {
        this.hoursWorked = hoursWorked;
    }
    @Override //The @Override annotation in Java is used to indicate that the
annotated method in a subclass is intended to override a method with the same
signature in its superclass. This annotation helps ensure that the method in
the subclass is correctly overriding a method in the superclass.  removing the
@Override annotation will not cause a compilation error, and your code will
still compile successfully.However, using @Override is a good practice because
it provides additional compile-time checks and helps prevent common mistakes.
    public double calculateWeeklySalary() {
        return hourlyRate * hoursWorked;
    }
    @Override
    public double calculateBonus() {
        return 0.05 * calculateWeeklySalary();
    }
}

class SalariedEmployee extends Employee {
    private double monthlySalary;

    public void setMonthlySalary(double monthlySalary) {
        this.monthlySalary = monthlySalary;
    }
    public double getMonthlySalary() {
        return monthlySalary;
    }
    @Override
    public double calculateWeeklySalary() {
        return monthlySalary / 4;
    }
    @Override
    public double calculateBonus() {
        return 0.1 * monthlySalary;
    }
}


class ExecutiveEmployee extends SalariedEmployee {
    private double bonusPercentage;

    public void setBonusPercentage(double bonusPercentage) {
        this.bonusPercentage = bonusPercentage;
    }
```

```
    @Override
    public double calculateBonus() {
        return super.calculateBonus() + (bonusPercentage / 100) *
getMonthlySalary();
    }
}

public class PayrollSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        HourlyEmployee hourlyEmployee = new HourlyEmployee();
        hourlyEmployee.setEmployeeId(777); // Employee Information ideally
sourced from web server databases.
        hourlyEmployee.setEmployeeName("Dr. Prabu P");
        hourlyEmployee.setDesignation("PG Professor");

        try {//Data Validation achieved with the implementation of try-catch
block
            System.out.print("Enter hourly rate: ");
            hourlyEmployee.setHourlyRate(scanner.nextDouble());
            System.out.print("Enter hours worked: ");
            hourlyEmployee.setHoursWorked(scanner.nextInt());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter a valid numeric
value.");
            return;
        }
        hourlyEmployee.displayDetails();
    }
}
```

Output :

```
PS C:\Users\rickyswas> F:
PS F:\> cd "Java"
PS F:\Java> cd "Lab3"
PS F:\Java\Lab3> javac PayrollSystem.java
PS F:\Java\Lab3> java PayrollSystem
Enter hourly rate: 500
Enter hours worked: 35
Employee ID: 777
Employee Name: Dr. Prabu P
Designation: PG Professor
Weekly Salary: 17500.0
Bonus: 875.0
PS F:\Java\Lab3>
```

Question 2 : PayrollSystemTest.java

Code :

```
import java.util.Scanner;

class Employee {
    private int employeeId;
    private String employeeName;
    private String designation;

    public void setEmployeeId(int employeeId) {
        this.employeeId = employeeId;
    }
    public void setEmployeeName(String employeeName) {
        this.employeeName = employeeName;
    }
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    public int getEmployeeId() {
        return employeeId;
    }
    public String getEmployeeName() {
        return employeeName;
    }
    public String getDesignation() {
        return designation;
    }
    public double calculateBonus() {// Default implementation. Can be
overridden in derived classes
        return 0.0;
    }
    public double calculateWeeklySalary() {// Default implementation. Can be
overridden in derived classes
        return 0.0;
    }
    public void displayDetails() {
        System.out.println("Employee ID: " + employeeId);
        System.out.println("Employee Name: " + employeeName);
        System.out.println("Designation: " + designation);
        System.out.println("Weekly Salary: " + calculateWeeklySalary());
        System.out.println("Bonus: " + calculateBonus());
    }
}

class HourlyEmployee extends Employee {
    private double hourlyRate;
    private int hoursWorked;
```

```
    public void setHourlyRate(double hourlyRate) {
        this.hourlyRate = hourlyRate;
    }
    public void setHoursWorked(int hoursWorked) {
        this.hoursWorked = hoursWorked;
    }
    @Override
    public double calculateWeeklySalary() {
        return hourlyRate * hoursWorked;
    }
    @Override
    public double calculateBonus() {
        return 0.05 * calculateWeeklySalary();
    }
}

class SalariedEmployee extends Employee {
    private double monthlySalary;

    public void setMonthlySalary(double monthlySalary) {
        this.monthlySalary = monthlySalary;
    }
    public double getMonthlySalary() {
        return monthlySalary;
    }
    @Override
    public double calculateWeeklySalary() {
        return monthlySalary / 4;
    }
    @Override
    public double calculateBonus() {
        // Bonus calculation specific to SalariedEmployee
        return 0.1 * monthlySalary;
    }
}

class ExecutiveEmployee extends SalariedEmployee {
    private double bonusPercentage;

    public void setBonusPercentage(double bonusPercentage) {
        this.bonusPercentage = bonusPercentage;
    }
    @Override
    public double calculateBonus() {
        // Bonus calculation specific to ExecutiveEmployee
        return super.calculateBonus() + (bonusPercentage / 100) *
getMonthlySalary();
```

```
    }
}

public class PayrollSystemTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter employee ID: ");
        int employeeId = 0;
        try {
            employeeId = scanner.nextInt();
        } catch (Exception e) {
        System.out.println("Invalid input. Please enter a valid numeric
employee ID.");
        return;
        }
        System.out.print("Enter employee name: ");
        scanner.nextLine();
        String employeeName = scanner.nextLine();
        System.out.print("Enter employee designation: ");
        String designation = scanner.nextLine();
        HourlyEmployee hourlyEmployee = new HourlyEmployee();// Test
HourlyEmployee
        hourlyEmployee.setEmployeeId(employeeId);
        hourlyEmployee.setEmployeeName(employeeName);
        hourlyEmployee.setDesignation(designation);
        try {
            System.out.print("Enter hourly rate: ");
            hourlyEmployee.setHourlyRate(scanner.nextDouble());

            System.out.print("Enter hours worked: ");
            hourlyEmployee.setHoursWorked(scanner.nextInt());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter valid numeric
values.");
            return;
        }
        SalariedEmployee salariedEmployee = new SalariedEmployee(); //Test
SalariedEmployee
        salariedEmployee.setEmployeeId(employeeId);
        salariedEmployee.setEmployeeName(employeeName);
        salariedEmployee.setDesignation(designation);
        try {
            System.out.print("Enter monthly salary: ");
            salariedEmployee.setMonthlySalary(scanner.nextDouble());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter a valid numeric
value.");
            return;
```

```
        }
        ExecutiveEmployee executiveEmployee = new ExecutiveEmployee(); //
Testing ExecutiveEmployee
        executiveEmployee.setEmployeeId(employeeId);
        executiveEmployee.setEmployeeName(employeeName);
        executiveEmployee.setDesignation(designation);
        try {
            System.out.print("Enter bonus percentage: ");
            executiveEmployee.setBonusPercentage(scanner.nextDouble());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter a valid numeric
value.");
            return;
        }
        System.out.println("Employee Details:");
        System.out.println("------------------");

        System.out.println("\nHourly Employee:");
        hourlyEmployee.displayDetails();
        displayAnnualEarnings(hourlyEmployee);

        System.out.println("\nSalaried Employee:");
        salariedEmployee.displayDetails();
        displayAnnualEarnings(salariedEmployee);

        // System.out.println("\nExecutive Employee:");
        // executiveEmployee.displayDetails();
        // displayAnnualEarnings(executiveEmployee);
        System.out.println("\nTotal Payroll:");

        double totalPayroll = calculateTotalPayroll(hourlyEmployee,
salariedEmployee, executiveEmployee);
        System.out.println("Total Payroll: Rs." + totalPayroll);
    }
    private static void displayAnnualEarnings(Employee employee) {
        System.out.println("Annual Earnings: Rs." +
calculateAnnualEarnings(employee));
    }
    private static double calculateAnnualEarnings(Employee employee) {
        return 52 * employee.calculateWeeklySalary();
    }
    private static double calculateTotalPayroll(Employee... employees) {
        double totalPayroll = 0.0;
        for (Employee employee : employees) {
            totalPayroll += calculateAnnualEarnings(employee);
        } return totalPayroll;
    }
}
```

Output :

```
PS F:\Java\Lab3> javac PayrollSystemTest.java
PS F:\Java\Lab3> java PayrollSystemTest
Enter employee ID: 777
Enter employee name: Dr. Prabu P
Enter employee designation: PG Professor
Enter hourly rate: 500
Enter hours worked: 35
Enter monthly salary: 100000
Enter bonus percentage: 10
Employee Details:
-----------------

Hourly Employee:
Employee ID: 777
Employee Name: Dr. Prabu P
Designation: PG Professor
Weekly Salary: 17500.0
Bonus: 875.0
Annual Earnings: Rs.910000.0

Salaried Employee:
Employee ID: 777
Employee Name: Dr. Prabu P
Designation: PG Professor
Weekly Salary: 25000.0
Bonus: 10000.0
Annual Earnings: Rs.1300000.0

Total Payroll:
Total Payroll: Rs.2210000.0
PS F:\Java\Lab3>
```