Practical-5.1 : Write a program which defines base class Employee having three data members, namely name [30], emp_numb and gender and two methods namely input_data() and show data(). Derive a class SalariedEmployee from Employee and add a new data member, namely salary. It also adds two member methods, namely allowance (if gender is female HRA=0.1 *salary else 0.09* salary. DA= 0.05*salary) and increment (salary= salary+0.1 *salary). Display the gross salary in the main class. (Tip: Use super to call base class's constructor).

Input:-

```
import java.util.Scanner;

class Employee {
    protected String name;
    protected int empNumber;
    protected char gender;

    public void input_data() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Employee Name: ");
        name = scanner.nextLine();
        System.out.print("Enter Employee Number: ");
        empNumber = scanner.nextInt();
        System.out.print("Enter Gender (M/F): ");
        gender = scanner.next().charAt(0);
    }

    public void showData() {
        System.out.println("Employee Name: " + name);
        System.out.println("Employee Number: " + empNumber);
        System.out.println("Gender: " + gender);
    }
}
class SalariedEmployee extends Employee {
    private double salary;

    public void input_data() {
        super.input_data();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Basic Salary: ");
        salary = scanner.nextDouble();
    }
    public double calculateAllowance() {
        double hra = (gender == 'F' || gender == 'f') ? 0.10 * salary : 0.09 * salary;
        double da = 0.05 * salary;
        return hra + da;
    }

    public void incrementSalary() {
        salary += 0.1 * salary;
    }

    public void displayGrossSalary() {
        showData();
        double grossSalary = salary + calculateAllowance();
        System.out.println("Basic Salary: " + salary);
        System.out.println("Allowance: " + calculateAllowance());
        System.out.println("Gross Salary: " + grossSalary);
        System.out.println("----------------------------");
    }
}

class Practical_51 {
    public static void main(String[] args) {
        SalariedEmployee emp = new SalariedEmployee();
        emp.input_data();
        System.out.println("\nBefore Increment:");
        emp.displayGrossSalary();
        emp.incrementSalary();
        System.out.println("\nAfter Increment:");
        emp.displayGrossSalary();
    }
}
```

Output:-

```
Enter Employee Name: abc
Enter Employee Number: 101
Enter Gender (M/F): m
Enter Basic Salary: 5000

Before Increment:
Employee Name: abc
Employee Number: 101
Gender: m
Basic Salary: 5000.0
Allowance: 700.0
Gross Salary: 5700.0
----------------------------

After Increment:
Employee Name: abc
Employee Number: 101
Gender: m
Basic Salary: 5500.0
Allowance: 770.0
Gross Salary: 6270.0
```

```
Enter Employee Name: anjali
Enter Employee Number: 102
Enter Gender (M/F): f
Enter Basic Salary: 5000

Before Increment:
Employee Name: anjali
Employee Number: 102
Gender: f
Basic Salary: 5000.0
Allowance: 750.0
Gross Salary: 5750.0
----------------------------

After Increment:
Employee Name: anjali
Employee Number: 102
Gender: f
Basic Salary: 5500.0
Allowance: 825.0
Gross Salary: 6325.0
```

Practical-5.2 : WAP that illustrates method overriding. Class A3 is extended by Class B3. Each of these classes defines a hello (string s) method that outputs the string "A3: Hello from" or "B3: Hello from" respectively. Use the concept Dynamic Method Dispatch and keyword super

Input:-

i.

```
class A3 {
    void hello(String s) {
        System.out.println("A3: Hello from " + s);
    }
}

class B3 extends A3 {
    void hello(String s) {
        super.hello(s);
        System.out.println("B3: Hello from " + s);
    }
}

public class MethodOverridingDemo {
    public static void main(String[] args) {
        A3 ref;
        B3 obj = new B3();
        ref = obj;
        ref.hello("Class B3");
    }
}
```

Output:-

```
A3: Hello from Class B3
B3: Hello from Class B3
```