1. Create class WageEmployee extending Employee class with attributes as hrs (int)and rate(int) and method computeSalary() to calculate the salary.Print the salary and details of WageEmployee. (Note: Use the previous Employee classes. Accept the values from the user..Default, Parameterised Constructor and toString() to be written in all the classes)

```
⇒ package com.prathamesh.jan27;
import java.util.Scanner;
class Employee {
    int hours;
    int rate;

    public void computeSalary() {
        System.out.println("The salary is" + (hours * rate));
    }
}

class WageEmploye extends Employee {
    int hrs;
    int rate;

    public void computeSalary() {
        System.out.println("The salary of WageEmployee is" + (hrs * rate));
    }

    public WageEmploye(int hrs, int rate) {
        this.hrs = hrs;
        this.rate = rate;
    }

    public String toString() {
        return "no of hrs are::" + hrs + " and rate is ::" + rate + "";
    }
}


public class Q_one {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int hrs = sc.nextInt();
        int rate = sc.nextInt();
        WageEmploye e1 = new WageEmploye(hrs, rate);
        e1.computeSalary();
        System.out.println(e1);
    }
}
```

Output:

```
8
550
The salary of WageEmployee is4400
no of hrs are::8and rate is ::550

Process finished with exit code 0
```

2. Create SalesPerson class extending WageEmployee with attributes as sales(int) and commission (int). Override the ComputeSalary() in Salesperson class and print the salary and details of SalesPerson.

⇒

```
package com.prathamesh.jan27;


class WageEmployee {
    int sales = 10;
    int commission = 10000;


    void ComputeSalary() {
        System.out.println("salary is ::" + (sales * commission));
    }
}

class SalesPerson extends WageEmployee {
    //Overiding Method
    public void ComputeSalary() {
        System.out.println("the sales is ::" + sales + " the commission is ::" + commission + "\n the " 
+
                "salary is ::" + (sales * commission));
    }
}

public class Q2 {
    public static void main(String[] args) {
        SalesPerson s1 = new SalesPerson();
        s1.ComputeSalary();
    }
}
```

Output:

```
the sales is ::10 the commission is ::10000
 the salary is ::100000

Process finished with exit code 0
```

3. Create Manager class extending Employee class with attributes as fixedsalary(int) and incentives(int) and method computeSalary() to calculate the salary of Manager .Print the salary and details of Manager

⇒ package com.prathamesh.jan27;

```
class WageEmployee {
    int fixedSalary  ;
    int incentive ;
    void ComputeSalary(){
        System.out.println("salary is ::"+(fixedSalary+incentive));
    }
}
class Manager extends WageEmployee {
    public void printData(){
        super.fixedSalary = 45000;
        super.incentive= 250;
        super.ComputeSalary();
    }
}
public class Q3 {
    public static void main(String[] args) {
        Manager m_one = new Manager();
        m_one.printData();
    }
}
```

Output:

```
salary is ::45250

Process finished with exit code 0
```

4. Write a TestEmployee class to print the details of all types of employees (use array[] of Employee class)

⇒

```
package com.prathamesh.Inheritance;


import java.util.Scanner;
```

```
class Employee {
    int hrs, rate;
    String name;

    Employee(int hrs, int rate, String name) {
        this.hrs = hrs;
        this.rate = rate;
        this.name = name;
    }

    Employee(String name) {
        this.name = name;
    }

    void computeSalary() {
        System.out.println("Salary is" + " " + hrs * rate);
    }

    public String toString() {
        return ("Hrs is: " + hrs + " " + "rate is: " + rate + " " + "Name is: " + name);
    }
}

class Manager extends Employee {
    int fixedSalary, incentives;

    Manager(String name, int fixedSalary, int incentives) {
        super(name);
        this.fixedSalary = fixedSalary;
        this.incentives = incentives;
    }

    @Override
    void computeSalary() {
        int salary = fixedSalary + incentives;
        System.out.println("Salary is: " + salary);
    }

    public String toString() {
        return "Name is: " + name + " " + "fixedSalary is: " + fixedSalary + " " + "Incentives are: " + incentives;
    }
}

class WageEmployee extends Employee {
    WageEmployee(int hrs, int rate, String name) {
        super(hrs, rate, name);
    }

    @Override
    void computeSalary() {
        System.out.println("Salary is" + " " + hrs * rate);
    }

    public String toString() {
        return ("Hrs is: " + hrs + " " + "rate is: " + rate + " " + "Name is: " + name);
    }
}

class SalesPerson extends WageEmployee {
    int sales, commission;


    SalesPerson(int hrs, int rate, String name, int sales, int commission) {
        super(hrs, rate, name);
        this.sales = sales;
        this.commission = commission;
    }

    @Override
    void computeSalary() {
        int salary = hrs * rate + sales + commission;
        System.out.println("Salary is" + " " + salary);
    }

    public String toString() {
        return "Hrs is: " + hrs + " " + "rate is: " + rate + " " + "Name is: " + name + " " + "sales is: " + sales + " " + "Commission is: " + commission;
    }
}

class TestEmployee {
    void print(Employee[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i].toString());
        }
    }
}

public class Ques_1 {
    public static void main(String[] args) {
        Scanner inp = new Scanner(System.in);
        System.out.println("How many Employees: ");
        int n = inp.nextInt();
        Employee[] arr = new Employee[n];
        for (int i = 0; i < n; i++) {
            System.out.println("manager/wage/sales ?");
            String category = inp.next();
            if (category.equals("manager")) {
                System.out.println("Enter name,fixedsalary,incentives");
                String name = inp.next();
                int fixedsalary = inp.nextInt();
                int incentives = inp.nextInt();
                Manager obj_m = new Manager(name, fixedsalary, incentives);
                arr[i] = obj_m;
                obj_m.computeSalary();
                System.out.println(obj_m.toString());
            } else if (category.equals("sales")) {
                System.out.println("Enter hrs,rate,name,sales,commission ");
                int hrs = inp.nextInt();
                int rate = inp.nextInt();
                String name = inp.next();
                int sales = inp.nextInt();
                int commission = inp.nextInt();
                SalesPerson obj_s = new SalesPerson(hrs, rate, name, sales, commission);
                arr[i] = obj_s;
                obj_s.computeSalary();
                System.out.println(obj_s.toString());
            } else {
                System.out.println("Enter hrs,rate,name");
                int hrs = inp.nextInt();
                int rate = inp.nextInt();
                String name = inp.next();
                WageEmployee obj_w = new WageEmployee(hrs, rate, name);
                arr[i] = obj_w;
                obj_w.computeSalary();
                ;
                System.out.println(obj_w.toString());
            }

        }
        TestEmployee obj_t = new TestEmployee();
        obj_t.print(arr);

    }
}
```

Output:-

```
How many Employees:
3
manager/wage/sales ?
manager
Enter name,fixedsalary,incentives
prath
11000
5000
Salary is: 16000
Name is: prath fixedSalary is: 11000 Incentives are: 5000
manager/wage/sales ?
wage
Enter hrs,rate,name
8
100
prathamesh
Salary is 800
Hrs is: 8 rate is: 100 Name is: prathamesh
manager/wage/sales ?
```

```
sales
Enter hrs,rate,name,sales,commission
8
200
Prathameshc
400000
5000
Salary is 406600
Hrs is: 8 rate is: 200 Name is: Prathameshc sales is: 400000 Commission is: 5000
Name is: prath fixedSalary is: 11000 Incentives are: 5000
Hrs is: 8 rate is: 100 Name is: prathamesh
Hrs is: 8 rate is: 200 Name is: Prathameshc sales is: 400000 Commission is: 5000

Process finished with exit code 0
```