

Name: Badal Wanjari

Branch: Computer Technology

Section: B

Roll No. 140

Registration No. 20011045

Subject: Object Oriented Programming Lab

## Practical-4

- **Problem Definition:**

Create a class named 'Member' having the following members.

Data members:

1. Name
2. Age
3. Phone number
4. Address
5. Salary

It also has a method named print in which prints the salary of members.

Two classes 'Employee' and 'Manager' inherit the member class the 'Employee' and 'Manager' classes having data members 'specialization' and 'department' respectively. Now design name, age, phone number, address and salary to an employee and a manager by making an object of both these classes and print the same using the override method 'printInfo'.

- **Program:**

```
class Member{
    String Name;
    int Age;
    String PhoneNo;
    String Address;
    Double Salary;

    //overriden method
    void printInfo() {
    }
}

class Employees extends Member {
```

```

String Specialization;

//constructor
public Employees(String name, int age, String address, String PhoneNum, double sal, String
specilizedIn) {
    this.Name = name;
    this.Age = age;
    this.Address = address;
    this.PhoneNo = PhoneNum;
    this.Salary = sal;
    this.Specialization = specilizedIn;
}

//overriding method
@Override
void printInfo() {
    System.out.println("Name : " + this.Name);
    System.out.println("Age : " + this.Age);
    System.out.println("Address : " + this.Address);
    System.out.println("Phone No. : " + this.PhoneNo);
    System.out.println("Post : Employee");
    System.out.println("Specialization : " + this.Specialization);
    System.out.println("Salary : " + "Rs. " + this.Salary + "\n");
}
}

class Manager extends Member {
    String Department;

    //constructor
    public Manager(String name, int age, String address, String PhoneNum, double sal, String
department) {
        this.Name = name;
        this.Age = age;
        this.Address = address;
        this.PhoneNo = PhoneNum;
        this.Salary = sal;
        this.Department = department;
    }

    //overriding method
    @Override
    void printInfo() {
        System.out.println("Name : " + this.Name);
        System.out.println("Age : " + this.Age);
        System.out.println("Address : " + this.Address);
        System.out.println("Phone No. : " + this.PhoneNo);
        System.out.println("Post : Manager");
        System.out.println("Department : " + this.Department);
        System.out.println("Salary : " + "Rs. " + this.Salary + "\n");
    }
}

```

```

    }
}

public class Practical4 {
    public static void main(String[] args) {
        Employees employee1 = new Employees("Dhiraj Deshmukh", 23, "Sakkardhara, Nagpur - 441100", "+919848684682", 30000, "Accounts");
        Manager manager1 = new Manager("Rajkumar Singh", 26, "Hingna, Nagpur - 441104", "+917856846820", 60000, "Sales");

        manager1.printInfo();
        employee1.printInfo();
    }
}

```

- **Output:**

Name : Rajkumar Singh  
 Age : 26  
 Address : Hingna, Nagpur - 441104  
 Phone No. : +917856846820  
 Post : Manager  
 Department : Sales  
 Salary : Rs. 60000.0

Name : Dhiraj Deshmukh  
 Age : 23  
 Address : Sakkardhara, Nagpur - 441100  
 Phone No. : +919848684682  
 Post : Employee  
 Specialization : Accounts  
 Salary : Rs. 30000.0

- **Screenshot:**

The screenshot displays the Visual Studio Code editor with a file named `Practical4.java` open. The code defines two classes: `Member` and `Employees`. The `Member` class has attributes `Name`, `Age`, `PhoneNo`, `Address`, and `Salary`, along with an overridden `printInfo()` method. The `Employees` class extends `Member` and adds a `Specialization` attribute, a constructor, and an overridden `printInfo()` method. The terminal on the right shows the execution of the code, displaying the output for two instances: `Rajkumar Singh` and `Dhiraj Deshmukh`.

```
1 class Member {
2     String Name;
3     int Age;
4     String PhoneNo;
5     String Address;
6     Double Salary;
7
8     //overridden method
9     void printInfo() {
10    }
11 }
12
13 class Employees extends Member {
14     String Specialization;
15
16     //constructor
17     public Employees(String name, int age, String address, String PhoneNum, double
18         this.Name = name;
19         this.Age = age;
20         this.Address = address;
21         this.PhoneNo = PhoneNum;
22         this.Salary = sal;
23         this.Specialization = specilizedIn;
24     }
25
26     //overriding method
27     @Override
28     void printInfo() {
29         System.out.println("Name : " + this.Name);
30         System.out.println("Age : " + this.Age);
31         System.out.println("Address : " + this.Address);
32         System.out.println("Phone No. : " + this.PhoneNo);
33         System.out.println("Post : " + this.Post);
34         System.out.println("Department : " + this.Department);
35         System.out.println("Salary : Rs. " + this.Salary);
36     }
37 }
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\Admin\Desktop\Notes\Data Structure Lab> cd "c:\Users\Admin\Desktop\Notes\OOP-Lab\" ; if (\$?) { javac Practical4.java } ; if (\$?) { java Practical4 }

Name : Rajkumar Singh  
Age : 26  
Address : Hingna, Nagpur - 441104  
Phone No. : +917856846820  
Post : Manager  
Department : Sales  
Salary : Rs. 60000.0

Name : Dhiraj Deshmukh  
Age : 23  
Address : Sakardhara, Nagpur - 441100  
Phone No. : +919848684682  
Post : Employee  
Specialization : Accounts  
Salary : Rs. 30000.0

PS C:\Users\Admin\Desktop\Notes\OOP-Lab>

- Result:**

By studying implementation of concept of run time polymorphism in Java., I have successfully completed Practical-4.