

# Inheritance, Overloading, Overriding

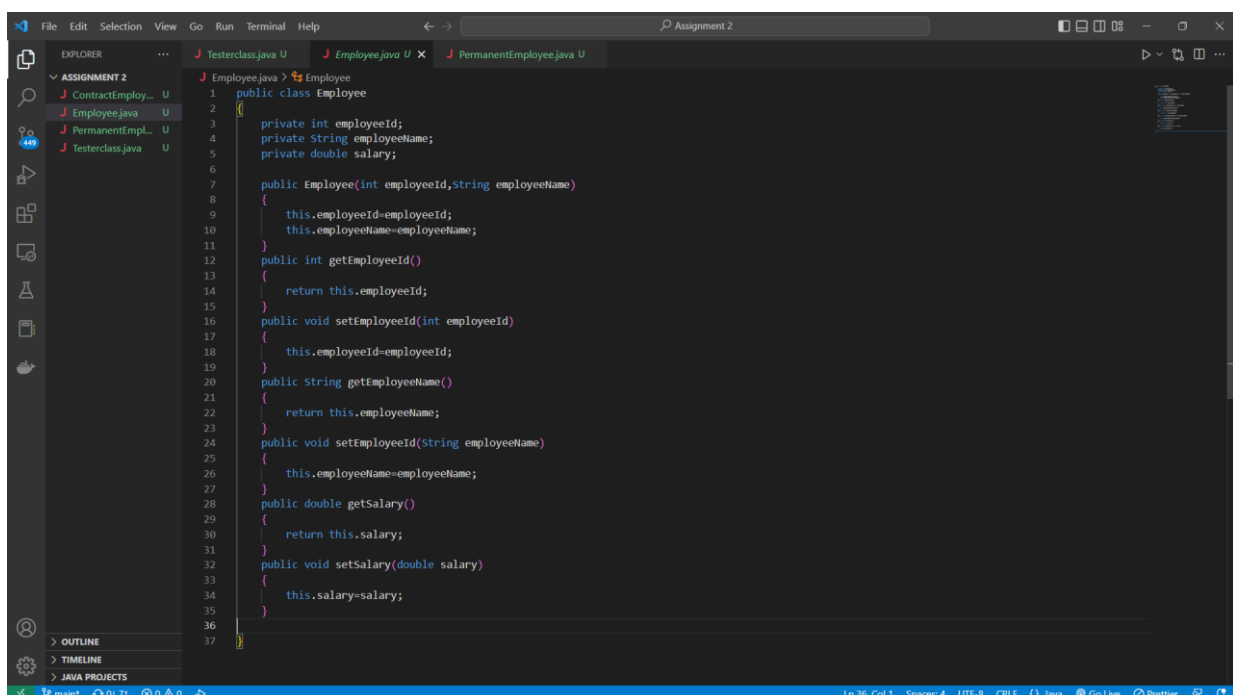
## Assignment 2

Name : Arijit Saha , College : Future Institute of Engineering and Management ,  
Employee Id : T23010330 , Mail Id: [arijits@trainee.nrifintech.com](mailto:arijits@trainee.nrifintech.com)

Q1. A construction company wants to keep a record of the employees working in it. There are permanent employees as well as contract employees. Contract employees work on an hourly basis whereas permanent employees are paid monthly salary. An application needs to be developed for the company for storing the employee details. Implement the classes based on the class diagram and description given below. Red Color -> indicates private Green Color-> indicates public

Method Description Employee Employee(int employeeId, String employeeName) Initialize the employeeId and employeeName instance variables appropriately with the values passed to the constructor. Implement the getter and setter methods appropriately. PermanentEmployee PermanentEmployee(int empId, String name, double basicPay, double hra, float experience) Initialize the employeeId, employeeName, basicPay, hra and experience instance variables appropriately with the values passed to the constructor. calculateMonthlySalary() Calculate and set the salary of the employee using the formula given below.  $\text{salary} = \text{basic pay} + \text{hra} + \text{variable component}$

## Employee.java



```
1 public class Employee
2 {
3     private int employeeId;
4     private String employeeName;
5     private double salary;
6
7     public Employee(int employeeId, String employeeName)
8     {
9         this.employeeId = employeeId;
10        this.employeeName = employeeName;
11    }
12    public int getEmployeeId()
13    {
14        return this.employeeId;
15    }
16    public void setEmployeeId(int employeeId)
17    {
18        this.employeeId = employeeId;
19    }
20    public String getEmployeeName()
21    {
22        return this.employeeName;
23    }
24    public void setEmployeeName(String employeeName)
25    {
26        this.employeeName = employeeName;
27    }
28    public double getSalary()
29    {
30        return this.salary;
31    }
32    public void setSalary(double salary)
33    {
34        this.salary = salary;
35    }
36 }
37
```

# ContractEmployee.java

```
1 public class ContractEmployee extends Employee{
2     private double wage;
3     private float hoursWorked;
4
5     public double getwage() {
6         return this.wage;
7     }
8
9     public void setwage(double wage) {
10        this.wage = wage;
11    }
12
13    public float gethoursWorked() {
14        return this.hoursWorked;
15    }
16
17    public void sethoursWorked(float hoursWorked) {
18        this.hoursWorked = hoursWorked;
19    }
20
21    public ContractEmployee(int employeeId, String employeeName, double wage, float hoursWorked) {
22        super(employeeId, employeeName);
23        this.wage=wage;
24        this.hoursWorked=hoursWorked;
25    }
26
27    void calculateSalary()
28    {
29        setSalary(this.hoursWorked*this.wage);
30    }
31 }
```

# PermanentEmployee.java

```
1 public class PermanentEmployee extends Employee {
2
3     private double basicPay;
4     private double hra;
5     private float experience;
6
7     public double getbasicPay() {
8         return this.basicPay;
9     }
10
11    public void setbasicPay(double basicPay) {
12        this.basicPay = basicPay;
13    }
14
15    public double gethra() {
16        return this.hra;
17    }
18
19    public void sethra(double hra) {
20        this.hra = hra;
21    }
22
23    public float getexperience() {
24        return this.experience;
25    }
26
27    public void setexperience(float experience) {
28        this.experience = experience;
29    }
30
31    public PermanentEmployee(int employeeId, String employeeName, double basicPay, double hra, float experience) {
32        super(employeeId, employeeName);
33    }
34 }
```

```
29 }
30
31 public PermanentEmployee(int employeeId, String employeeName, double basicPay, double hra, float experience) {
32     super(employeeId, employeeName);
33     this.basicPay = basicPay;
34     this.hra = hra;
35     this.experience = experience;
36 }
37
38 void calculateMonthlySalary()
39 {
40     double variableComponent = 0;
41     float experience = this.experience;
42     if (experience < 3)
43     {
44         variableComponent = 0;
45     }
46     else if (experience >= 3 && experience < 5)
47     {
48         variableComponent += (basicPay * 5) / 100;
49     }
50     else if (experience >= 5 && experience < 10)
51     {
52         variableComponent += (basicPay * 7) / 100;
53     }
54     else if (experience >= 10)
55     {
56         variableComponent += (basicPay * 12) / 100;
57     }
58     double sal = Math.round(this.basicPay + this.hra + variableComponent);
59     setSalary(sal);
60 }
```

## Testerclass.java Along with Output

```
1 public class Testerclass {
2     public static void main(String[] args) {
3         ContractEmployee c1 = new ContractEmployee(employeeId: 101, employeeName: "Akash", wage: 200, hoursWorked: 5);
4         PermanentEmployee p1 = new PermanentEmployee(employeeId: 301, employeeName: "Jit", basicPay: 50000, hra: 5500, experience: 15);
5         c1.calculateSalary();
6         p1.calculateMonthlySalary();
7
8         System.out.println("Salary of " + c1.getEmployeeName() + " is : " + c1.getSalary());
9         System.out.println("Salary of " + p1.getEmployeeName() + " is : " + p1.getSalary());
10    }
11 }
12
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ARIJIT\Desktop\WRI\Assignments\WEEK 2\Inheritance, Overloading, Overriding\Assignment 2> c: cd 'c:\Users\ARIJIT\Desktop\WRI\Assignments\WEEK 2\Inheritance, Overloading, Overriding\Assignment 2'; & 'c:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ARIJIT\AppData\Roaming\Code\User\workspaceStorage\3a97abcd37d32cc6045f4bb00a66a7b\redhat.java\jdk\_ws\Assignment 2\_e30702f7\bin' 'Testerclass'

Salary of Akash is : 1000.0  
Salary of Jit is : 61500.0

PS C:\Users\ARIJIT\Desktop\WRI\Assignments\WEEK 2\Inheritance, Overloading, Overriding\Assignment 2> []