

QUESTION 01

```
J MostFrequentNumbers.java  J Vehicle.java  J main.java 2  J Employee.java X  J Matrix.java
Java Lecture > src > J Employee.java > Main > main(String[])
1  // Base class representing an employee
2  public abstract class Employee {
3      private int employeeId;
4      private String employeeName;
5      private String designation;
6
7      public Employee(int employeeId, String employeeName, String designation) {
8          this.employeeId = employeeId;
9          this.employeeName = employeeName;
10         this.designation = designation;
11     }
12
13     public int getEmployeeId() {
14         return employeeId;
15     }
16
17     public void setEmployeeId(int employeeId) {
18         this.employeeId = employeeId;
19     }
20
21     public String getEmployeeName() {
22         return employeeName;
23     }
24
25     public void setEmployeeName(String employeeName) {
26         this.employeeName = employeeName;
27     }
28
29     public String getDesignation() {
30         return designation;
31     }
32
33     public void setDesignation(String designation) {
34         this.designation = designation;
35     }
36
37     public abstract double calculateWeeklySalary();
38 }
```

```
J MostFrequentNumbers.java J Vehicle.java J main.java 2 J Employee.java X J Matrix.java J University.java J ArrayProcessor.java J
Java Lecture > src > J Employee.java > Employee > getEmployeeName()
37 public abstract double calculateWeeklySalary();
38
39 public abstract void displayEmployeeDetails();
40
41 public abstract double calculateBonus();
42 // Method to calculate the annual earnings, combining weekly salary and bonus
43 public double calculateAnnualEarnings() {
44     return calculateWeeklySalary() * 52 + calculateBonus();
45 }
46 }
47
48 class HourlyEmployee extends Employee {
49     private double hourlyRate;
50     private int hoursWorked;
51
52     public HourlyEmployee(int employeeId, String employeeName, String designation, double hourlyRate, int hoursWorked) {
53         super(employeeId, employeeName, designation);
54         this.hourlyRate = hourlyRate;
55         this.hoursWorked = hoursWorked;
56     }
57
58     public double getHourlyRate() {
59         return hourlyRate;
60     }
61
62     public void setHourlyRate(double hourlyRate) {
63         this.hourlyRate = hourlyRate;
64     }
65
66     public int getHoursWorked() {
67         return hoursWorked;
68     }
69
70     public void setHoursWorked(int hoursWorked) {
71         this.hoursWorked = hoursWorked;
72     }
73 // Method to calculate the weekly salary for an hourly employee
```

```
J MostFrequentNumbers.java J Vehicle.java J main.java 2 J Employee.java X J Matrix.java J University.java J
Java Lecture > src > J Employee.java > HourlyEmployee > getHourlyRate()
73 // Method to calculate the weekly salary for an hourly employee
74 @Override
75 public double calculateWeeklySalary() {
76     return hourlyRate * hoursWorked;
77 }
78
79 @Override
80 public void displayEmployeeDetails() {
81     System.out.println("Employee ID: " + getEmployeeId());
82     System.out.println("Employee Name: " + getEmployeeName());
83     System.out.println("Designation: " + getDesignation());
84     System.out.println("Hourly Rate: " + getHourlyRate());
85     System.out.println("Hours Worked: " + getHoursWorked());
86     System.out.println("Weekly Salary: " + calculateWeeklySalary());
87 }
88 // Method to calculate the bonus for an hourly employee
89 @Override
90 public double calculateBonus() {
91     return 0; // No bonus for hourly employees
92 }
93 }
94 // Class representing a salaried employee, extending the Employee class
95 class SalariedEmployee extends Employee {
96     private double monthlySalary;
97
98     public SalariedEmployee(int employeeId, String employeeName, String designation, double monthlySalary) {
99         super(employeeId, employeeName, designation);
100         this.monthlySalary = monthlySalary;
101     }
102
103     public double getMonthlySalary() {
104         return monthlySalary;
105     }
106
107     public void setMonthlySalary(double monthlySalary) {
108         this.monthlySalary = monthlySalary;
109     }
110 }
```

```

110
111     @Override
112     public double calculateWeeklySalary() {
113         return monthlySalary / 4;
114     }
115
116     @Override
117     public void displayEmployeeDetails() {
118         System.out.println("Employee ID: " + getEmployeeId());
119         System.out.println("Employee Name: " + getEmployeeName());
120         System.out.println("Designation: " + getDesignation());
121         System.out.println("Monthly Salary: " + getMonthlySalary());
122         System.out.println("Weekly Salary: " + calculateWeeklySalary());
123     }
124     // Method to calculate the bonus for a salaried employee
125     @Override
126     public double calculateBonus() {
127         return monthlySalary * 0.1; // 10% of monthly salary as bonus
128     }
129 }
130
131 // Class representing an executive employee, extending the SalariedEmployee class
132 class ExecutiveEmployee extends SalariedEmployee {
133     private double bonusPercentage;
134
135     public ExecutiveEmployee(int employeeId, String employeeName, String designation, double monthlySalary, double bonusPercentage) {
136         super(employeeId, employeeName, designation, monthlySalary);
137         this.bonusPercentage = bonusPercentage;
138     }
139
140     public double getBonusPercentage() {
141         return bonusPercentage;
142     }
143
144     public void setBonusPercentage(double bonusPercentage) {
145         this.bonusPercentage = bonusPercentage;
146     }
147     // Method to calculate the bonus for an executive employee
148 }

```

```

149 // Main class for testing
150 class Main {
151     public static void main(String[] args) {
152         Employee hourlyEmployee = new HourlyEmployee(employeeId:1, employeeName:"prabu", designation:"Hourly Employee", hourlyRate:10, hoursWorked:40);
153         hourlyEmployee.displayEmployeeDetails();
154         System.out.println("Annual Earnings: " + hourlyEmployee.calculateAnnualEarnings());
155
156         Employee salariedEmployee = new SalariedEmployee(employeeId:2, employeeName:"senthinathan", designation:"Salaried Employee", monthlySalary:5000);
157         salariedEmployee.displayEmployeeDetails();
158         System.out.println("Annual Earnings: " + salariedEmployee.calculateAnnualEarnings());
159
160         Employee executiveEmployee = new ExecutiveEmployee(employeeId:3, employeeName:"helen", designation:"Executive Employee", monthlySalary:8000, bonusPercentage:0.15);
161         executiveEmployee.displayEmployeeDetails();
162         System.out.println("Annual Earnings: " + executiveEmployee.calculateAnnualEarnings());
163     }
164 }

```

```
Employee ID: 1
Employee Name: prabu
Designation: Hourly Employee
Hourly Rate: 10.0
Hours Worked: 40
Weekly Salary: 400.0
Annual Earnings: 20800.0
Employee ID: 2
Employee Name: senthilnathan
Designation: Salaried Employee
Monthly Salary: 5000.0
Weekly Salary: 1250.0
Annual Earnings: 65500.0
Employee ID: 3
Employee Name: helen
Designation: Executive Employee
Monthly Salary: 8000.0
Weekly Salary: 2030.0
Annual Earnings: 105680.0
PS D:\Java Projects>
```

QUESTION 02

```
J MostFrequentNumbers.java  J Vehicle.java  J main.java 2  J Employee.java  J PayrollTest.java X  J Matrix.java  J University.java  J ArrayProcessor.java  J BankAccount.java 6  J Share

Java Lecture > src > J PayrollTest.java > PayrollTest > main(String[])
1 public class PayrollTest {
2     public static void main(String[] args) {
3         // Test HourlyEmployee
4         HourlyEmployee hourlyEmployee = new HourlyEmployee(employeeId:1, employeeName:"prabu", designation:"Hourly Worker", hourlyRate:15.0, hoursWorked:40);
5         hourlyEmployee.displayEmployeeDetails();
6         System.out.println("Weekly Salary: " + hourlyEmployee.calculateWeeklySalary());
7         System.out.println("Bonus: " + hourlyEmployee.calculateBonus());
8         System.out.println("Annual Earnings: " + hourlyEmployee.calculateAnnualEarnings());
9         System.out.println();
10
11        // Test SalariedEmployee
12        SalariedEmployee salariedEmployee = new SalariedEmployee(employeeId:2, employeeName:"senthilnathan", designation:"Manager", monthlySalary:6000.0);
13        salariedEmployee.displayEmployeeDetails();
14        System.out.println("Weekly Salary: " + salariedEmployee.calculateWeeklySalary());
15        System.out.println("Bonus: " + salariedEmployee.calculateBonus());
16        System.out.println("Annual Earnings: " + salariedEmployee.calculateAnnualEarnings());
17        System.out.println();
18
19        // Test ExecutiveEmployee
20        ExecutiveEmployee executiveEmployee = new ExecutiveEmployee(employeeId:3, employeeName:"helen", designation:"Executive", monthlySalary:10000.0, bonusPercentage:0.15);
21        executiveEmployee.displayEmployeeDetails();
22        System.out.println("Weekly Salary: " + executiveEmployee.calculateWeeklySalary());
23        System.out.println("Bonus: " + executiveEmployee.calculateBonus());
24        System.out.println("Annual Earnings: " + executiveEmployee.calculateAnnualEarnings());
25        System.out.println();
26
27        // Display total payroll
28        displayTotalPayroll(hourlyEmployee, salariedEmployee, executiveEmployee);
29    }
30
31    // Display total payroll for all employees
32    private static void displayTotalPayroll(Employee... employees) {
33        double totalPayroll = 0;
34        for (Employee employee : employees) {
35            totalPayroll += employee.calculateAnnualEarnings();
36        }
37        System.out.println("Total Payroll: " + totalPayroll);
38    }
39 }
40
```

```
Employee ID: 1
Employee Name: prabu
Designation: Hourly Worker
Hourly Rate: 15.0
Hours Worked: 40
Weekly Salary: 600.0
Weekly Salary: 600.0
Bonus: 0.0
Annual Earnings: 31200.0

Employee ID: 2
Employee Name: senthilnathan
Designation: Manager
Monthly Salary: 6000.0
Weekly Salary: 1500.0
Weekly Salary: 1500.0
Bonus: 600.0
Annual Earnings: 78600.0

Employee ID: 3
Employee Name: helen
Designation: Executive
Monthly Salary: 10000.0
Weekly Salary: 2537.5
Weekly Salary: 2537.5
Bonus: 150.0
Annual Earnings: 132100.0

Total Payroll: 241900.0
PS D:\Java Projects>
```