

## Programming Test Results (With Test Cases)

### Result Summary

Field	Value
Test ID	39149
Student ID	29195
Programs (with test cases)	1
Total Test Cases	3
Test Cases Passed	3
Fully Passed Programs	1
Partially Passed Programs	0
Failed Programs	0
Overall % (with test cases)	100.00%
Grade	Outstanding

### Programs With Test Cases

#	Program Name	Total TC	Passed	Success Rate	Score	Submitted At	Attempts
1	EmployeeSystem	3	3	100.0%	10	12/11/2025, 12:30:17	0

### Program Details (With Test Cases)

#### Program 1: EmployeeSystem

Field	Value
Program No.	1
Program Name	EmployeeSystem
Description	<p>Create a Program to Develop Hierarchical Inheritance for Employee Management System</p> <p>Here Employee is super class and it has two sub classes PermanentEmployee and ContractEmployee.</p> <p>We need to display the details of PermanentEmployee including provident fund and net Salary.</p> <p>We need to display the details of ContractEmployee including contract duration.</p> <p>Coding Requirements :</p> <p>-----</p>

Field	Value
	<p>Create a BLC class called Employee.</p> <p>Attributes/ Properties [protected]      employeeId - int      employeeName - String      employeeSalary - double</p> <p>Create a Parameterized constructor to initialize all fields.</p> <p>Create another BLC class PermanentEmployee which extends from Employee</p> <p>Attributes/ Properties [protected]      providentfund - double</p> <p>Create a parameterized constructor to initialize super class properties only</p> <p>[Note : providentfund property will not initialize through constructor]</p> <p>Initialize providentfund manually, It must be 12% of the salary.</p> <p>Generate toString() method to print all the properties of super class as well as sub class.  [See the test cases for more details]</p> <p>Method :</p> <p>-----</p> <p>Method Name : netSalary()  Modifier : public  Argument : No argument  return type : void</p> <p>This method should print netSalary i.e salary + providentfund</p> <p>Create another BLC class ContractEmployee which extends from Employee</p> <p>Attributes/ Properties [protected]</p> <p>contractDuration - int</p> <p>Create a parameterized constructor to initialized all fields (super class and sub class)</p> <p>Generate toString() method to print all the properties of super class as well as sub class.  [See the test cases for more details]</p> <p>Create an ELC class EmployeeSystem which contains main method.</p> <p>Create both sub class objects and print the result.</p>
Constraints	
Sample Input	101 John 50000 102 Mike 30000 12
Sample Output	PermanentEmployee [employeeId=101, employeeName=John, employeeSalary=50000.0, providentfund=6000.0] Net Salary: 56000.0 ContractEmployee [employeeId=102, employeeName=Mike, employeeSalary=30000.0, contractDuration=12]

Field	Value
Explanation	-
Language(s)	java
Total Test Cases	3
Test Cases Passed	3
Test Cases Failed	0
Success Rate	100.0%
Score (0–10)	10
Attempts	0
Submitted At	12/11/2025, 12:30:17

Code

```

import java.util.*;
public class EmployeeSystem{
    public static void main(String [ ] args){
        Scanner sc = new Scanner(System.in);
        int emid = Integer.parseInt(sc.nextLine());
        String name = sc.nextLine();
        double salary = Double.parseDouble(sc.nextLine());

        if(salary<0){
            System.out.println("Error Invalid Input");
            System.exit(0);
        }
        else{
            PermanentEmployee p = new PermanentEmployee(emid,name,salary);
            System.out.println(p.toString());

            int id2 = Integer.parseInt(sc.nextLine());
            String name2 = sc.nextLine();
            double sala =Double.parseDouble(sc.nextLine());
            int dur = Integer.parseInt(sc.nextLine());

            ContractEmployee c = new ContractEmployee(id2,name2,sala,dur);
            System.out.println(c.toString());
        }
    }
}

class Employee{
    protected int empId;
    protected String empName;
    protected double empSalary;

    Employee(int empId, String empName, double empSalary){
        this.empId=empId;
        this.empName=empName;
        this.empSalary = empSalary;
    }
}

class PermanentEmployee extends Employee{
    protected double providedfund;
    PermanentEmployee(int empId, String empName, double empSalary){
        super(empId,empName,empSalary);
        this.providedfund =(empSalary*0.12);
    }
    public String toString(){
        return "PermanentEmployee [employeeId="+empId+", employeeName="+empName+",\nemployeeSalary="+empSalary+", providentfund="+providedfund+"] "+"\\n"+ "Net Salary:\n"+(empSalary+providedfund);
    }
}

class ContractEmployee extends Employee{
    protected int contractDuration;
    ContractEmployee(int empId, String empName, double empSalary, int contractDuration){
        super(empId,empName,empSalary);
        this.contractDuration=contractDuration;
    }
    public String toString(){
        return "ContractEmployee [employeeId="+empId+", employeeName="+empName+,\nemployeeSalary="+empSalary+", contractDuration="+contractDuration+"]";
    }
}

```