**Assignment No.**: 4  
**Assignment Name**: Designing a System with Inheritance

**Java Program: Maintaining Information System with Inheritance**

We will design a system to maintain information about **Employees** and **Managers** in a company. The system will use inheritance, where the Manager class will inherit from the Employee class. Both classes will have attributes and methods to display and manage the employee and manager information.

// Base class: Employee

class Employee {

private String name;

private int employeeId;

private double salary;

// Default constructor

public Employee() {

this.name = "Unknown";

this.employeeId = 0;

this.salary = 0.0;

}

// Parameterized constructor

public Employee(String name, int employeeId, double salary) {

this.name = name;

this.employeeId = employeeId;

this.salary = salary;

}

// Getter and setter methods

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getEmployeeId() {

return employeeId;

}

public void setEmployeeId(int employeeId) {

this.employeeId = employeeId;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

// Method to display employee details

public void displayEmployeeInfo() {

System.out.println("Employee Name: " + name);

System.out.println("Employee ID: " + employeeId);

System.out.println("Employee Salary: $" + salary);

}

}

// Derived class: Manager (inherits from Employee)

class Manager extends Employee {

private String department;

private double bonus;

// Default constructor

public Manager() {

super(); // Calls the constructor of Employee

this.department = "Unknown";

this.bonus = 0.0;

}

// Parameterized constructor

public Manager(String name, int employeeId, double salary, String department, double bonus) {

super(name, employeeId, salary); // Calls the parameterized constructor of Employee

this.department = department;

this.bonus = bonus;

}

// Getter and setter methods

public String getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

public double getBonus() {

return bonus;

}

public void setBonus(double bonus) {

this.bonus = bonus;

}

// Method to display manager details (overrides the display method of Employee)

@Override

public void displayEmployeeInfo() {

super.displayEmployeeInfo(); // Call base class method

System.out.println("Manager Department: " + department);

System.out.println("Manager Bonus: $" + bonus);

}

}

// Main class to test the system

public class CompanyInfoSystem {

public static void main(String[] args) {

// Creating an Employee object

Employee employee1 = new Employee("John Doe", 101, 50000);

System.out.println("Employee 1 Details:");

employee1.displayEmployeeInfo();

// Creating a Manager object

Manager manager1 = new Manager("Alice Smith", 201, 75000, "IT", 15000);

System.out.println("\nManager 1 Details:");

manager1.displayEmployeeInfo();

// Creating another Manager object

Manager manager2 = new Manager("Bob Johnson", 202, 80000, "HR", 20000);

System.out.println("\nManager 2 Details:");

manager2.displayEmployeeInfo();

}

}

**Output:**

Employee 1 Details:

Employee Name: John Doe

Employee ID: 101

Employee Salary: $50000.0

Manager 1 Details:

Employee Name: Alice Smith

Employee ID: 201

Employee Salary: $75000.0

Manager Department: IT

Manager Bonus: $15000.0

Manager 2 Details:

Employee Name: Bob Johnson

Employee ID: 202

Employee Salary: $80000.0

Manager Department: HR

Manager Bonus: $20000.0