

Title : Write a program for implementing multilevel inheritance for employee class.

Aim : To demonstrate the concept of multilevel inheritance.

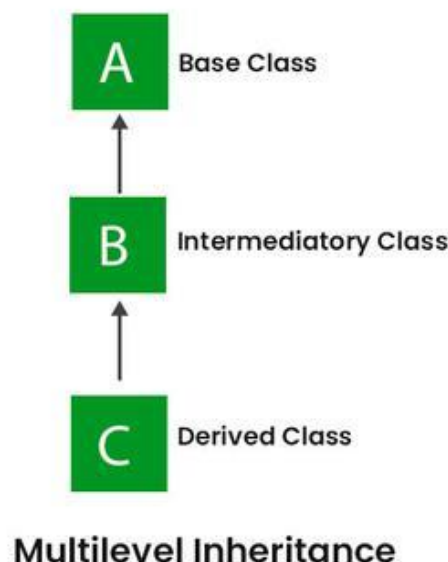
Objective : To learn the structure of multilevel inheritance.

Theory :

Multilevel inheritance in Java refers to the inheritance hierarchy where a derived class extends another derived class, forming a chain of inheritance. In other words, it involves inheriting from a class that is already inheriting from another class.

Note that the classes must be at different levels. Hence, there exists a single base class and single derived class but multiple intermediate base classes.

Example :



As shown in above block diagram, class C has class B and class A as parent classes. Depending on the relation the level of inheritance can be extended to any level. As in other inheritance, based on the visibility mode used or access specifier used while

deriving, the properties of the base class are derived. Access specifier can be private, protected or public.

Procedure :

```
// Person.java
```

```
class Person {  
  
    String name;  
  
    int age;  
  
    public Person(String name, int age) {  
  
        this.name = name;  
  
        this.age = age;  
  
    }  
  
    public void displayDetails() {  
  
        System.out.println("Name: " + name);  
  
        System.out.println("Age: " + age);  
  
    }  
  
}
```

```
// Employee.java (Child of Person)
```

```
class Employee extends Person {  
  
    String employeeId;  
  
    double salary;  
  
    public Employee(String name, int age, String employeeId, double salary) {  
  
        super(name, age);  
  
        this.employeeId = employeeId;  
  
        this.salary = salary;  
  
    }  
  
    public void displayEmployeeDetails() {  
  
        displayDetails();  
  
    }  
  
}
```

```

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

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

    }
}

// Manager.java (Child of Employee)
class Manager extends Employee {

    String department;

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

        super(name, age, employeeId, salary);

        this.department = department;

    }

    public void displayManagerDetails() {

        displayEmployeeDetails();

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

    }

}

public class Main {

    public static void main(String[] args) {

        Manager manager = new Manager("Your Name", 25, "PRN", 50000, "Engg");

        manager.displayManagerDetails();

    }

}

```