ASSIGNMENT NO. 02.

PROBLEM STATEMENT:
Write a program in Java showing hierarchical inheritance with base class as Employee and derived classes as FullTimeEmployee and InternEmployee with methods DisplaySalary in base class and CalculateSalary in derived classes. Calculate salary method will calculate as per increment given to fullTime and intern Employees. FullTime employee - 50% hike, Intern employee - 20% hike. Display salary before and after hike.

OBJECTIVE:
1. To study inheritance in Java.
2. To study why to use Inheritance.
3. To study types of Inheritance.

THEORY:

1. What is inheritance in Java?
→ Inheritance is an OOP concept where a new class (child/subclass) acquires the properties and methods of an existing class (parent/superclass).

2. Why use inheritance?
→ -Promotes code reusability
-Reduces duplication
-Improves maintainability.
-Supports method overriding.

Types in Inheritance.
1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance
4. Multiple Inheritance (using interfaces)
5. Hybrid Inheritance.

ALGORITHM/CLASS DIAGRAM/IMPLEMENTATION:

Algorithm:
1. Start
2. Create base class Employee with variable salary.
3. Define method displaySalary() in base class.
4. Create subclass ~~InternEmployee~~ FullTimeEmploy extending Employee.
5. Create subclass InternEmployee extending Employ
6. Define calculateSalary() in both subclasses.
7. Accept salary input.
8. Display salary before hike
9. Apply hike based on employee type.
10. Display salary after hike.
11. Stop.

CONCLUSION: Thus, we have successfully implemented hierarchical inheritance in Java, where multiple ~~classe~~ subclasses inherit from a single base class. This demonstrates code reuse and specialization of behavior.

FAQs :

1. Is multiple inheritance supported in Java? How is it achieved?

→ No, Java does not support multiple inheritance using classes. It is achieved using interfaces.

2. What is IS-A relationship in Java?

→ Inheritance represents an IS-A relationship.
EX - InternEmployee is an Employee.

3. Are constructor and instance initialisation block inherited to subclass?

No, constructors and instance initialization blocks are not inherited, but constructors of the superclass are called using super().