INHERITANCE

1. What is Inheritance?

Inheritance is a mechanism in Java that allows one class to acquire the properties (variables) and behaviors (methods) of another class. It is based on the "is-a" relationship. For example, a "Dog" is-a "Animal."

Key Terms:

- Superclass (Parent): The class whose features are being inherited.
- Subclass (Child): The class that inherits the features. It can reuse the parent's code and add its own unique features.

General Term	Parent Side	Child Side
Common Java Terms	Superclass	Subclass
Logical Relationship	Parent Class	Child Class
Structural Relationship	Base Class	Derived Class

Daily Life Example: Family Relationship

Think of Inheritance like how children inherit traits from their parents.

- Parent Class (Superclass) → Father/Mother
- Child Class (Subclass) → Son/Daughter

For example:

- Parents have common traits like surname, eye color, height.
- Children inherit these traits automatically, but they can also have their own unique features (like hobbies or talents).

2. The extends Keyword

To create an inheritance relationship, you use the extends keyword. This tells Java that one class is derived from another.

Simple Example:

```
// Parent Class
class Vehicle {
   int speed = 60;

   void drive() {
      System.out.println("The vehicle is moving.");
   }
}

// Child Class
class Car extends Vehicle {
   int doors = 4;

   void honk() {
      System.out.println("Beep Beep!");
   }
}
```

In this example, a Car object will have access to both speed and the drive() method, even though they aren't written inside the Car class.

3. Important Concepts for Beginners

- The super Keyword:

  The super keyword is used to refer directly to the parent class. It is commonly used for:

  Calling the parent's constructor: Ensuring the parent part of the object is set up first.

  Accessing parent methods: If a child has a method with the same name as the parent, super.methodName() allows you to call the parent's version.

- Method Overriding:

  This is when a subclass provides its own specific version of a method that already exists in the parent class. This allows the child to change how a parent behavior works.

- Constructor Chaining

  In Java, when you create an instance of a child class, the parent class constructor is automatically called first. This ensures that the base "blueprint" is built before the specific details are added.

4. Types of Inheritance in Java

- Single Inheritance: One class extends one other class (e.g., B extends A).
- Multilevel Inheritance: A chain of inheritance (e.g., C extends B, and B extends A).
- Hierarchical Inheritance: Multiple classes extend the same parent (e.g., B extends A, and C extends A).

Note: Java does not support "Multiple Inheritance" with classes (one child having two parents) because it causes confusion in the code's logic.

5. Benefits of Inheritance

- Code Reusability: You write common code once in the parent class and reuse it in many child classes.
- Organization: It helps group related objects together logically.
- Maintenance: Changes made to a parent class automatically update all child classes.