

6. Java Inheritance

Tech Tips
With Ajlok

• Inheritance in Java :

- Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented Programming System).
- The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from the existing class, you can reuse methods and fields of the parent class. Moreover, you can add methods and fields in your current class also.
- Inheritance represents the Is-A-relationship which is also known as a parent-child relationship.

➤ Why use inheritance in Java ?

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Syntax :

```
class Subclass-name extends Superclass-name  
{  
    // methods and fields  
}
```

Example :

```
class Doctor {
    float salary = 50000;
}

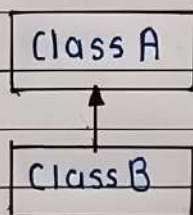
class Surgeon extends Doctor {
    float bonus = 20000;
    public static void main (String[] args) {
        Surgeon s = new Surgeon();
        System.out.println("Salary of Surgeon:" + s.salary);
        System.out.println("bonus of Surgeon:" + s.bonus);
    }
}
```

Output :

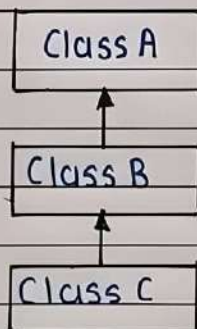
Salary of Surgeon: 50000.0
bonus of Surgeon: 20000.0

• Types of Inheritance in Java

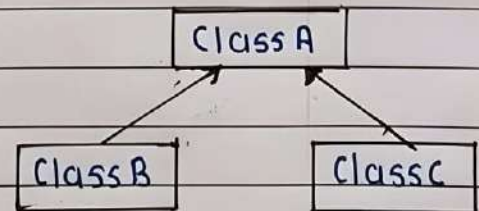
- On the basis of class, there can be three types of inheritance in java : Single, multilevel and hierarchical.
- In java programming, multiple and hybrid inheritance is supported through interface only. We will learn about interfaces later.



1) Single



2) Multilevel

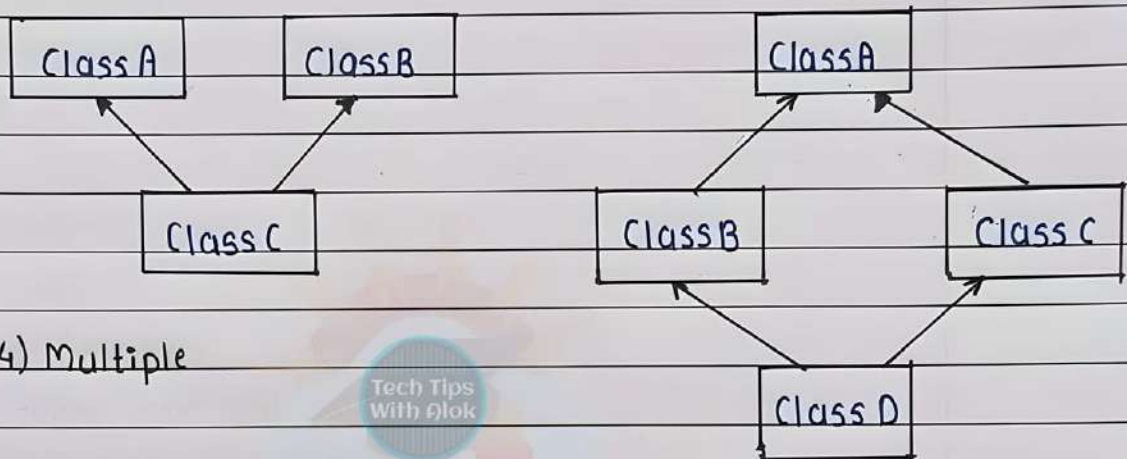


3) Hierarchical



Note : Multiple inheritance is not supported in java through class.

When one class inherits multiple classes, it is known as multiple inheritance. For Example :



1) Single Inheritance -

When a class inherits another class, it is known as single inheritance.

Example -

```
class Human
{
    void work()
    {
        System.out.println("Human can work");
    }
    void eat()
    {
        System.out.println("Human can eat");
    }
}
```



```
public class Engineer extends Human
{
    public static void main(String[] args)
    {
        Engineer E = new Engineer();
        E.work();
        E.eat();
    }
}
```

Output :

Human can work

Human can eat

Tech Tips
With Glok

2) Multilevel Inheritance -

- When there is a chain of inheritance is known as multilevel inheritance.
- In other words, one class extending another class and again another class extending the subclass is called as multilevel inheritance.

Example :

```
class Human
{
    void work()
    {
        System.out.println("Human can work");
    }
}
```

```
class Engineer extends Human {  
    void eat()  
    {  
        System.out.println("Engineer can eat");  
    }  
}  
  
public class Student extends Engineer  
{  
    public static void main(String[] args)  
    {  
        Student s = new Student();  
        s.work();  
        s.eat();  
    }  
}
```

Tech Tips
With Alok

Output :

Human can work
Engineer can eat.

3) Hierarchical Inheritance -

When two or more classes inherit a single class, it is known as hierarchical inheritance.

Example :

```
class Human {  
    void work()  
    {  
        System.out.println("Human can work");  
    }  
}
```



```
void eat()
{
    System.out.println("Human can eat");
}

class Engineer extends Human
{
}

public class Doctor extends Human
{
    public static void main(String[] args) {
        Doctor D = new Doctor();
        Engineer E = new Engineer();
        D.work();
        E.eat();
    }
}
```

Tech Tips
With @lok

Output:

Human can work

Human can eat

• Super Keyword in Java :

The 'super' keyword in java is used to refer to the superclass and its members from within the subclass. It is often used to access overridden methods, constructor chaining, and to differentiate between superclass, subclass members with the same time.



Example :

```
class Animal {  
    void eat() { System.out.println("eating..."); }  
}  
  
class Dog extends Animal {  
    void eat() { System.out.println("eating bread..."); }  
    void bark() { System.out.println("barking..."); }  
    void work() {  
        super.eat();  
        bark();  
    }  
}  
  
class TestSuper {  
    public static void main(String[] args) {  
        Dog d = new Dog();  
        d.work();  
    }  
}
```

Tech Tips
With Alok

• Aggregation in Java :

- If a class have an entity reference, it is known as Aggregation. Aggregation represents HAS-A relationship.

Example :

```
public class Address {  
    String city, state, country;  
  
    public Address(String city, String state, String country) {
```