

# Java Inheritance with



## ***INHERITANCE-***

*Inheritance is an important pillar of Object- Oriented Programing.*

*Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours of parent objects.*

*Java Inheritance is also called a concept of SubClass And SuperClass.*

*Inheritance represents the IS-A relationship which is aslo known as a parent-child relationship.*

## ***Why Use Inheritance in Java-***

### ***For Code Reusability-***

*Inheritance enables code Reusability and Saves Time. In place of writing the same code again and again we can simply inherit the properties of one class into other.*

### ***For Method Overriding-***

*A subclass provides the specific implementation of method that has been already declared by one of its parent class.*

*Method Overriding is used for runtime polymorphism.*

## ***Keywords Terminologies In Inheritance-***

***Extends***-- To Inherit from a class, use the extends keyword.

***Super***-- This is used to call the method of the parent class from the method of the child class.

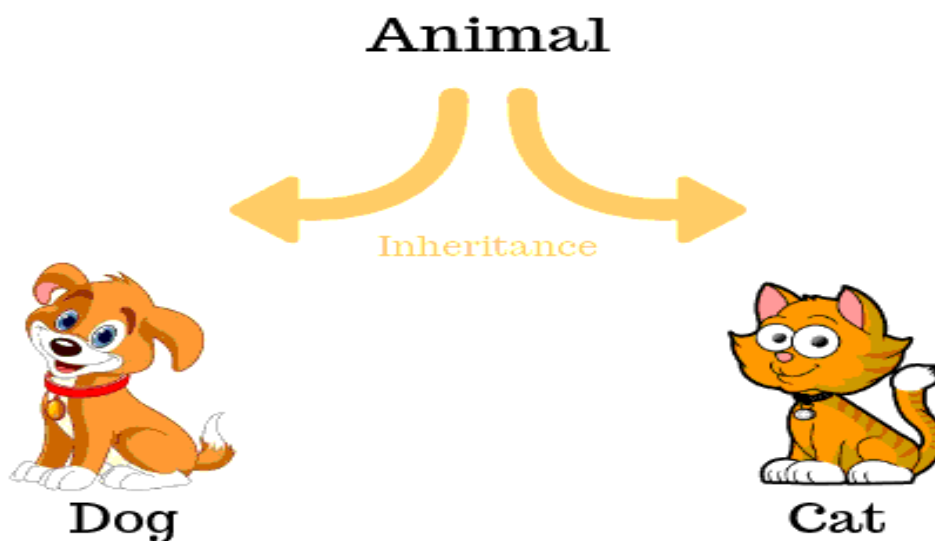
***Subclass***-- The class That Inherits from another class ie child.

***Superclass***-- The class Being Inherited from ie Parent.

## ***Syntax for Inheritance-***

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

## ***Inheritance Example-***



*Here, “Animal” is a class ie superclass and it has two subclasses - class “Dog” and “Cat” respectively. Class “Animal” may have some its own properties or behaviours. And the subclasses may also have some own properties and behavior. But the Concept of Inheritance says that the class Dog and Cat inherits the superclass Animal. Or alternatively we can say the superclass being inherited from the subclasses Dog and Cat.*

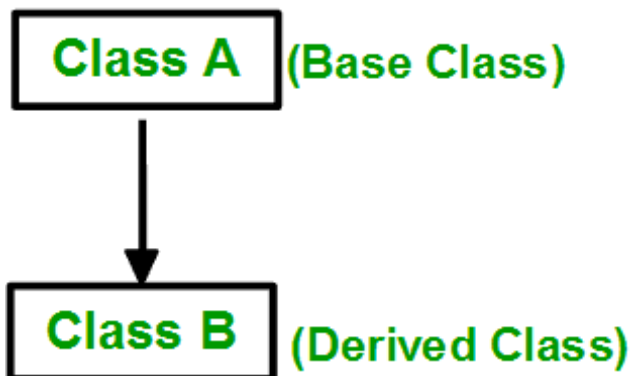
*That means Dog and Cat classes might copy the properties and behaviour of class Animal.*

## Types Of Inheritance in JAVA-

*Single Inheritance*

*Multilevel Inheritance*

*Hierarchical Inheritance*



### *Single Inheritance-*

*When a class inherits another class.*

```
import java.util.*;
class animal{
    public void eat( ){
        System.out.println(" They eat to live");
    }
    public void wander( ){
        System.out.println("They use to wander in search of food");
    }
}
class fox extends animal {
    public void aboutfox( ){
        super.eat( );
        System.out.println("fox are non-vegaterien");
        System.out.println("fox are very fast at running");
    }
}

public class main{
    public static void main(String[ ] args )
```

```

{
fox f1 = new fox( );
f1.aboutfox( )
f1.eat( );
f1.wander( );
}
}

```

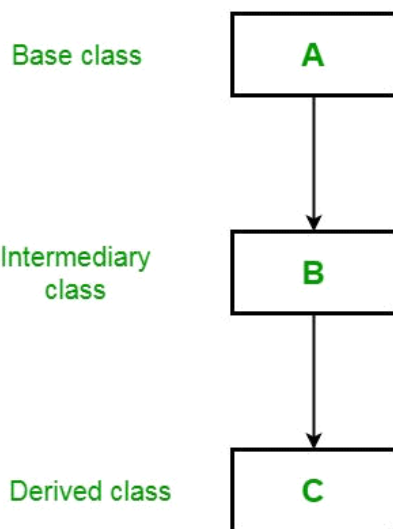
## OUTPUT-

*they it to live*  
*fox are non-vegaterien*  
*fox are very fast at running*  
*They eat to live*  
*They use to wander in search of food*

## Multilevel Ineritance-

*A derived class will be inheriting a base class and as well as the derived class aslo act as the base class to other class.*

```
class plusMinus{
```



```
void add(int x, int y){
```

```
System.out.println("arithmetic operations");
```

```
}
```

```
void add(int x, int y){
```

```
int z1=x+y;
```

```
System.out.println("addition is " + z1);
```

```

    }
    void sub(int x, int y){
        int z2=x-y;
        System.out.println("subtraction is " + z2);
    }
}
class mulDiv extends plusMinus{
    void product(int x, int y){
        int z3= x*y;
        System.out.println("multiplication is " + z3);
    }
    void divide(int x, int y){
        int z4= x/y;
        System.out.println("division is " + z4);
    }
}
class calculation extends mulDiv{
    void cal(){
        System.out.println("calculating....");
    }
}
class multiple_inheritance{
    public static void main(String[] args) {
        calculation c = new calculation();
        mulDiv m = new muldiv();
        m.arith();
    }
}

```

```
        c.cal();
        c.add(3,7);
        c.sub(7,3);
        c.product(8,5);
        c.divide(20,5);
    }
}
```

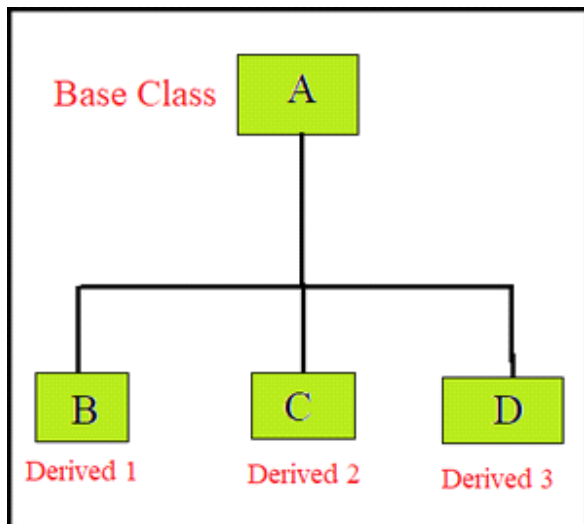
### **OUTPUT-**

```
arithmetic operations
calculating....
addition is10
subtraction is4
multiplication is40
division is4
```

Here, class "mulDiv" is accessing the methods of class "plusMinus" and class "calculation" is accessing the methods of class "mulDiv" and "plusMinus" both. This is the advantage of multilevel inheritance.

### **Hierarchical Inheritance-**

When two or more classes inherit a single class, it is known as hierarchical inheritance. There is a one superclass(base class) for more than one subclass.



```

}import java.util.*;

class bank{
    public void security( ){
        System.out.println(" bank is useful for secuiry of wealth and insuranance
of life");
    }
    public void interest( ){
        System.out.println("bank is giving interests for every cosumer");
    }
}

class AXIX extends bank {
    public void axis_bank( ){
        System.out.println("AXIS is giving interest at 5% yearly");
    }
}

class SBI extends bank {
    public void sbi_bank( ){
        System.out.println(" SBI is giving interest at 3.5% yearly");
    }
}

class BOB extends bank {
    public void bob_bank( ){
        System.out.println(" BOB is giving interest at 2.7% yearly");
    }
}

public class main{
    public static void main(String[ ] args )
    {
        AXIX a = new AXIX( );
        BOB b = new BOB( );
    }
}

```

```
SBI s = new SBI();
a.security();
b.interest();
b.bob_bank();
s.SBI_bank();
s.interest();
}
}
```

### **OUTPUT-**

*bank is useful for security of wealth and insurance of life*

*bank is giving interests for every consumer*

*BOB is giving interest at 2.7% yearly*

*SBI is giving interest at 3.5% yearly*

*bank is giving interests for every consumer*

*To Reduce the Complexity and Simplify the Language, Multiple inheritance is not supported in JAVA. Hybrid inheritance is also not possible with classes in JAVA.*

*These may be implemented through interfaces*