

Name: Shaikh Salif Aminuddin

CLASS: S2 **ROLL NO:** 2201094

EXPERIMENT NO: 8

AIM: Programs on types of inheritance.

THEORY:

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also. Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order. Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours of a parent object. Inheritance represents the IS-A relationship which is also known as a parentchild relationship. Why use inheritance in java

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

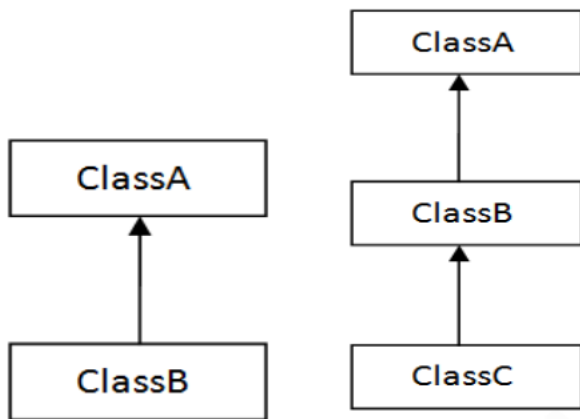
Terms used in Inheritance

- o Class: A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- o Sub Class/Child Class: Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- o Super Class/Parent Class: Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- o Reusability: As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

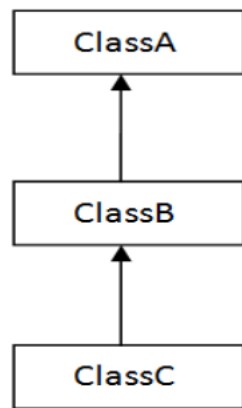
The syntax of Java Inheritance

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

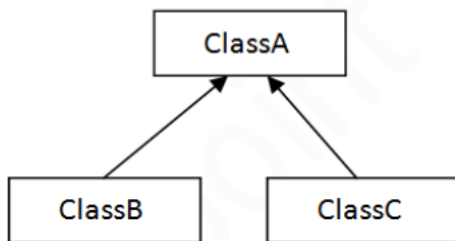
The extends keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality. Types of inheritance in java On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.



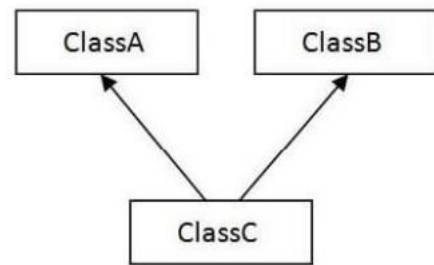
1) Single



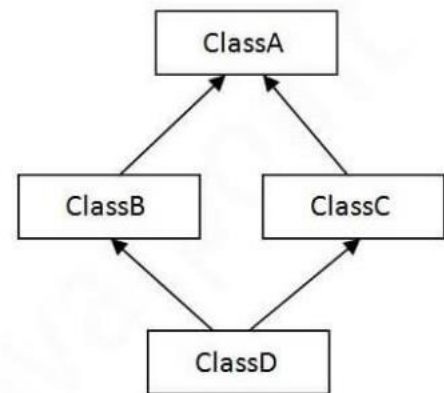
2) Multilevel



3) Hierarchical



4) Multiple



5) Hybrid

CODE: Single

```

class A {
    void add(int a,int b) {
        System.out.println("Addition is" +(a+b));
    }
}
class B extends A {
    void sub(int a,int b) {
        System.out.println("Subtraction is" +(a-b));
    }
}
class Inherit {
    public static void main(String args[]) {
        B obj1=new B();
        obj1.add(10,20);
        obj1.sub(20,10)
    }
}
  
```

OUTPUT:

```
C:\Users\15L>javac Inherit.java

C:\Users\15L>java Inherit
Addition is30
Subtraction is10
```

CODE: Multilevel

```
class A {
    void add(int a,int b) {
        System.out.println("Addition is" +(a+b));
    }
}
class B extends A {
    void sub(int a,int b) {
        System.out.println("Subtraction is" +(a-b));
    }
}
class C extends B {
    void mul(int a,int b) {
        System.out.println("Multiplication is" +(a*b));
    }
}
class Inherit {
    public static void main(String args[]) {
        C obj1=new C();
        Obj1.add(5,10);
        obj1.sub(10,3);
        obj1.mul(20,20);
    }
}
```

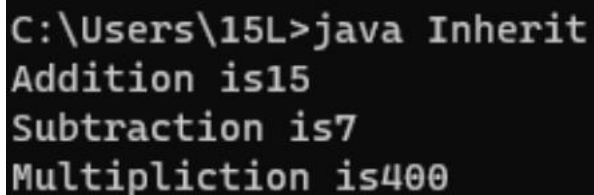
OUTPUT:

```
C:\Users\15L>java Inherit
Addition is15
Subtraction is7
Multiplication is400
```

CODE: Hierarchical

```
class A {  
    void add(int a,int b) {  
        System.out.println("Addition is" +(a+b));  
    }  
}  
class B extends A {  
    void sub(int a,int b) {  
        System.out.println("Subtraction is" +(a-b));  
    }  
}  
class C extends B {  
    void mul(int a,int b) {  
        System.out.println("Multiplication is" +(a*b));  
    }  
}  
class D extends B {  
    void div(int a,int b) {  
        System.out.println("Division is" +(a/b));  
    }  
}  
class Inherit {  
    public static void main(String args[]) {  
        C obj1=new C();  
        D obj2=new D();  
        obj2.add(5,10);  
        obj1.sub(10,3);  
        obj2.mul(20,20);  
    }  
}
```

OUTPUT:



```
C:\Users\15L>java Inherit  
Addition is15  
Subtraction is7  
Multiplication is400
```

CONCLUSION: In conclusion, inheritance is a fundamental concept in Java and object-oriented programming that allows for the creation of hierarchical relationships between classes. It enables the creation of new classes based on existing ones, promoting code reusability and a more organized and efficient code structure.