

Q) what is Inheritance and describe the types of Inheritance

A) Inheritance:

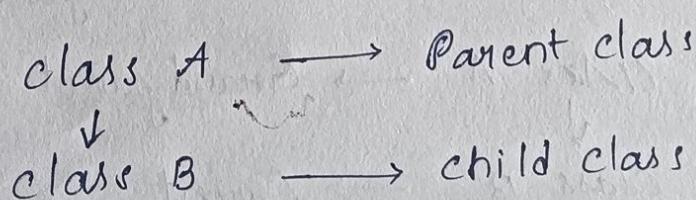
The method to create a hierarchy between classes by inheriting from other classes.

They are 5 types of Inheritance:-

- They are :-
- 1) Single Inheritance
  - 2) Multi level Inheritance
  - 3) Multiple Inheritance
  - 4) Hierarchical Inheritance
  - 5) Hybrid Inheritance

In this Multiple Inheritance doesn't supports in java to overcome we use Interface:-

1) Single Inheritance:- In this we have parent class and child class, Both are interlinked and child class accessed by the parent class.



Program:-

```
class A {  
    public void display {
```

```
        System.out.println("Basic class is Derived");  
    }  
}  
Class B extends A {  
    Public void dis_b() {  
        System.out.println("child class is created");  
    }  
}
```

```
Class Main {
```

```
    Public static void main(String[] args) {
```

```
        b.obj = new B();
```

```
        obj.dis_a();
```

```
        obj.dis_b();
```

Output: Basic class is  
Derived  
child class is created

2) Multi level Inheritance:- In this we have Grand Parent, Parent and child classes where child class becomes -Grand child for the Grand parent class.

Class A → Grand parent



Class B → Parent



Class C → Grand child / child

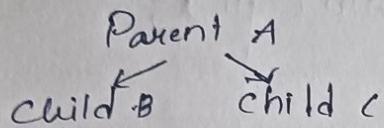
### Program:

```
class A{  
    public void dis1(){  
        System.out.println("Hi");  
    }  
}  
  
class B extends A{  
    public void dis2(){  
        System.out.println("Hello");  
    }  
}  
  
class C extends B{  
    public void dis3(){  
        System.out.println("world");  
    }  
}  
  
class Main{  
    public static void main(String[] args){  
        C o = new();  
        o.dis1();  
        o.dis2();  
        o.dis3();  
    }  
}
```

Output: Hi  
Hello  
world.

### 3) Hierarchical Inheritance:

In this Inheritance single parent class have many child classes as follows.



#### Program:-

class A {

```
    public void dis1() {
```

```
        System.out.println("Parent class derived");
```

```
}
```

```
}
```

class B extends A {

```
    public void dis2() {
```

```
        System.out.println("Child is derived");
```

```
}
```

```
}
```

class C extends A {

```
    public void dis3() {
```

```
        System.out.println("Child2 is derived");
```

```
}
```

class Main {

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

```
        B o1 = new B();
```

```
        C o2 = new C();
```

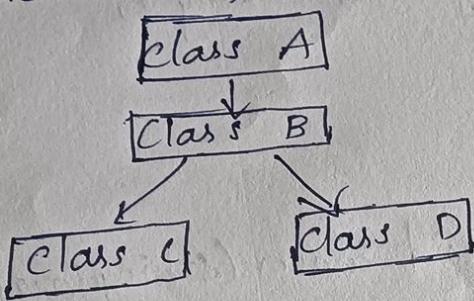
```
01. dis1();  
01. dis2();  
02. dis3();
```

Output :- Parent class derived  
child1 is derived  
child2 is derived

}

#### 4) Hybrid Inheritance:

In this inheritance is done when combination of 2 inheritances as follows:-



#### Program:-

```
class a {
```

```
    public void dis1() {  
        System.out.println("Hi");  
    }
```

Class b extends a {

```
    public void dis2() {  
        System.out.println("Hello");  
    }
```

}

class c extends B {

```
    public void dis3() {  
        System.out.println("Hi!");
```

}

class d extends B {

```
    public void dis4() {  
        System.out.println("King");
```

}

class Main{

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

```
        C objc = new C();
```

```
        D objd = new D();
```

```
        objc.dis1();
```

```
        objc.dis2();
```

```
        objc.dis3();
```

```
        objd.dis1();
```

```
        objd.dis2();
```

```
        objd.dis3();
```

Output:-

Hi

Hello

Hi!

Hi!

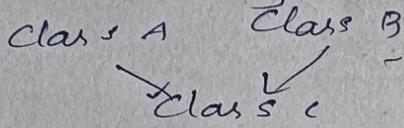
Hello

King

## 5) Multiple Inheritance:

In this Inheritance 2 parent class combines and forms a single child where in java it is not

Possible for over coming we use interface.



Class A {

int a;

A() {

a = 5;

Void dis1() {

System.out.println(a);

}

}

int b = 10;

Void dis2() {

System.out.println(b);

}

Class C extends A implements B {

int c = 15;

Void dis3() {

System.out.println(c);

}

Public class maind

```
public static void main(String[] args) {  
    obj=new C();  
    obj.du1();  
    obj.du2();  
    obj.du3();  
}  
  
3) class main {  
    public static void main(String[] args) {  
        try {  
            int a=50/0;  
            System.out.println("There is no errors");  
        }  
        catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
        finally {  
            System.out.println("This is finally Block");  
        }  
    }  
}  
output:- / by zero  
this is the finally block:
```