**IInheritanceINHERITANCE:**

1. **The main purpose of inheritance is code reusability…**
2. **Creating a new class from an existing class is known as**

**inheritance…**

1. **The existing class is known as Base class or parent or super class…**
2. **The newly created class is known as Derived class or Child or**

**sub class…**

1. **When we inherit a new class, the existing class members can be accessed in newly created class based on their accessibility levels…**

**TYPES OF INHERITANCE:**

**\* Single Inheritance**

**\* Multiple Inheritance**

**\* Multi-level Inheritance**

**SINGLE INHERITANCE:**

 **Creating a single new class from a single Existing class is known as single**

**Inheritance…**

**Class A Parent**

**Class B Child**

class Demo1

{

public void Show()

{

System.out.println("This is from show in Demo1");

}

}

class Demo2 extends Demo1

{

public void Test()

{

System.out.println("This is from Test in Demo2");

}

}

class Demo

{

public static void main(String args[])

{

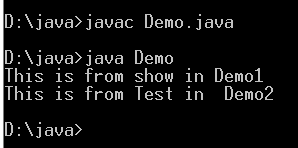
Demo2 obj=new Demo2();

obj.Show();

obj.Test();

}

}



2.Multilevel inheritance

A

B

C

3.Multiple Inheritance:

A show()

C show() show()

B show()

Obj.show()

C extends A,B----Error

Super keyword

class Demo1

{

int a;

public Demo1(int x)

{

a=x;

}

public void Show()

{

System.out.println("a value "+a);

}

}

class Demo2 extends Demo1

{

int b;

public Demo2(int x,int y)

{

super(x);

b=y;

}

public void Show1()

{

System.out.println("b value "+b);

}

}

class Demo

{

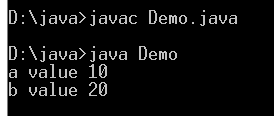
public static void main(String args[])

{

Demo2 abc=new Demo2(10,20);

abc.Show();

abc.Show1();

}}

class Demo1

{

int a;

public Demo1(int x)

{

a=x;

}

public void Show()

{

System.out.println("a value "+a);

}

}

class Demo2 extends Demo1

{

int b;

public Demo2(int x,int y)

{

super(x);

b=y;

}

public void Show1()

{

System.out.println("a value "+super.a);

}

}

class Demo

{

public static void main(String args[])

{

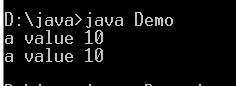
Demo2 abc=new Demo2(10,20);

abc.Show();

abc.Show1();

}

}



Final keyword:

final class Demo1

{

int a;

public Demo1(int x)

{

a=x;

}

public void Show()

{

System.out.println("a value "+a);

}

}

class Demo2 extends Demo1

{

int b;

public Demo2(int x,int y)

{

super(x);

b=y;

}

public void Show1()

{

System.out.println("b value "+b);

}

}

class Demo

{

public static void main(String args[])

{

Demo2 abc=new Demo2(10,20);

abc.Show();

abc.Show1();

}

}

