**Inheritance**

**Has\_A Examples**

class SuperA{

int x=111; }

class SubB {

int y = 222;

SuperA sa = new SuperA( ); }

class Has\_AEx1{

public static void main(String args[])

{ SubB sb=new SubB( );

System.out.println("y val is : "+sb.y);

System.out.println("Sa val is : "+sb.sa);

System.out.println("x val is : "+sb.sa.x); }

}

**Example 2:**

class Master{

void print()

{ System.out.println("Hello "); }

}

class Systems{

static Master out=new Master(); }

class Has\_AEx2{

public static void main(String args[])

{ Systems.out.print(); }

}

**Is\_A Relationship Examples :**

class SuperA{

private int x=100;

int y=200;

static int z=300; }

class SubB extends SuperA {

int a=400; }

class Is\_AEx1{

public static void main(String args[])

{ SubB sb=new SubB( );

// System.out.println("x val is : "+sb.x); CE

System.out.println("y val is : "+sb.y);

System.out.println("z val is : "+SubB.z);

System.out.println("z val is : "+SuperA.z);

System.out.println("z val is : "+sb.z);

System.out.println("a val is : "+sb.a);

SuperA sa=new SuperA( );

// System.out.println("x val is : "+sa.x);CE

}

}

**Example 2:**

class SuperA{

int x=111;

}

class SubB extends SuperA{

int x=222;

void method1()

{ int x=333;

System.out.println("x val is : "+x);

System.out.println("x val is : "+this.x);

System.out.println("x val is : "+super.x); }

public static void main(String args[])

{ SubB sb=new SubB( );

sb.method1(); }

}

**Example 3:**

class SuperA{

void method1()

{System.out.println("M1 of SA "); }

}

class SubB extends SuperA{

void method1()

{ super.method1();

System.out.println("M1 of SB "); }

public static void main(String args[])

{ SubB sb=new SubB( );

// super.method1();

sb.method1();

}

}

**Constructors In Inheritance :**

class SuperA{

SuperA()

{System.out.println("Def SA "); }

}

class SubB extends SuperA{

SubB()

{System.out.println("Def of SB "); }

public static void main(String args[])

{ SubB sb=new SubB( ); }

}

**Example 2:**

class SuperA{

SuperA()

{System.out.println("Def SA "); }

SuperA(int x)

{this();

System.out.println("Para of SA : "+x); }

}

class SubB extends SuperA{

SubB()

{ super(123);

System.out.println("Def of SB "); }

public static void main(String args[])

{ SubB sb=new SubB( ); }

}

**Example 3:**

class SuperA{

SuperA(int x)

{System.out.println("para of SA "+x);}

}

class SubB extends SuperA{

SubB(int x)

{ super(x);

System.out.println("para of SB : "+x); }

public static void main(String args[])

{ SubB sb=new SubB(123);}

}

**Abstract Methods And Abstract Classes**

abstract class Shapes

{ float dim1,dim2;

void setShapes(float dim1,float dim2)

{ this.dim1=dim1; this.dim2=dim2; }

abstract float findArea();

}

class Rect extends Shapes{

float findArea()

{ return (dim1\*dim2); }

}

class Triangle extends Shapes{

float findArea()

{ return (0.5f\*dim1\*dim2);}

}

class AbstractDemo{

public static void main(String args[])

{Shapes s=new Rect();

s.setShapes(4.0f,4.0f);

float ar=s.findArea( );

System.out.println("Area of Rect : "+ar);

s=new Triangle();

s.setShapes(5.0f,5.0f);

float at=s.findArea();

System.out.println("Area of Tri : "+at);

}

}

**Interface Example**

interface IA{

public abstract void method1();

}

class SubB implements IA{

public void method1()

{ System.out.println("OR m1 of IA"); }

public static void main(String args[])

{ IA ia=new SubB();

ia.method1(); }

}

**Ex 2:**

interface IA {

void method1(); }

interface IB extends IA{

void method2();

}

class SubB implements IB{

public void method1()

{System.out.println("OR m1 of IA "); }

public void method2()

{System.out.println("OR m2 of IB "); }

public static void main(String args[])

{ IA a=new SubB();

a.method1();

// a.method2(); CE

IB b=new SubB();

b.method2();

b.method1();

}

}

**Eg 3 :**

interface IA{ void method1(); }

interface IB{ void method2(); }

class SubB implements IA,IB

{ public void method1()

{System.out.println("M1 of IA ");}

public void method2()

{System.out.println("M2 of IB "); }

public static void main(String args[])

{ IA a=new SubB();

a.method1();

//a.method2(); CE

IB b=new SubB();

b.method2();

// b.method1(); CE

}

}

**Final Methods :**

**Eg 1:**

class SuperA{

final void method1()

{ System.out.println("m1 of SA "); }

}

class SubB extends SuperA

{ }

class FinalMtdEx1{

public static void main(String args[])

{ SubB sb=new SubB( );

sb.method1(); }

}

**Eg 2:**

class SuperA{

final void method1()

{ System.out.println("m1 of SA "); }

}

class SubB extends SuperA

{ void method1()

{}

}

class FinalMtdEx1{

public static void main(String args[])

{ SubB sb=new SubB( );

sb.method1(); }

}

**Final Classes :**

Eg 1:

final class A{

int x=111;

void method1()

{System.out.println("M1 of A : "+x); }

}

class Test{

public static void main(String args[])

{ A a=new A( );

a.method1(); }

}

Eg 2:

final class A{}

class B extends A{ }