**Interface in Java**

An interface in Java is a blueprint of a class. A Java interface contains static constants and abstract methods.

The interface in Java is *a* mechanism to achieve **abstraction**. There can be **only abstract methods and constant fields** in the Java interface, **not the method body**. It is used to **achieve abstraction** and **multiple inheritance in Java.**

Syntax:

interface {

// declare constant fields

// declare methods that abstract

// by default.

}

## **Why use Java interface?**

There are mainly three reasons to use interface. They are given below.

* It is used to achieve abstraction.
* By interface, we can support the functionality of multiple inheritance.
* It can be used to achieve loose coupling (low dependency among different classes)

**// defining interface**

interface Shape {

double Area(int x, int y);

}

// defining Rect class with implementation of Shape interface (I.e defining Area())

class Rect implements Shape{

int length,breadth;

public double Area(int l, int b)

{

length=l;

breadth=b;

return length\*breadth;

}

}

// defining Tria class with implementation of Shape interface (I.e defining Area())

class Tria implements Shape{

int breadth,height;

public double Area(int l, int b)

{

breadth=b;

height=l;

return 0.5\*breadth\*height;

}

}

// defining Square class with implementation of Shape interface (I.e defining Area())

class Square implements Shape{

int side;

public double Area(int l, int b)

{

side=b;

return side\*side;

}

}

public class InterfaceExample {

public static void main (String[] args)

{

// creating object of Interface

Shape s;

// creating an object for the class Rect

Rect r= new Rect();

// copying the rectangle object into interface Shape object s

s=r;

// As the s is the copy of the rectangle class , s.Area() will print area of rectangle

System.out.println(" The area of rectangle is "+s.Area(5,8));

// similarly for other triangle and square also

Tria t= new Tria();

s=t;

System.out.println(" The area of triangle is "+s.Area(9, 7));

Square sq =new Square();

s=sq;

System.out.println(" The area of square is "+s.Area(7,7));

}

}

o/p

The area of rectangle is 40.0

The area of triangle is 31.5

The area of square is 49.0

**Implementation of multiple inheritance using interface**

Rect class

Tria class

Square class

Shape

Interface with abstract method Area()