**Interface in Java**

An interface in Java is a blueprint of a class. It has static constants and abstract methods.

**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.

**How to declare an interface?**

An interface is declared by using the interface keyword.

**Syntax:**

interface <interface\_name>{

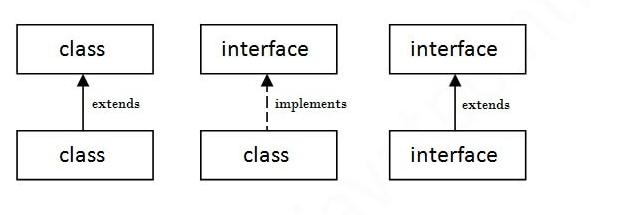
// declare constant fields

// declare methods that abstract

// by default.

}

**The relationship between classes and interfaces**

****

**Java Interface Example**

interface printable{

void print();

}

class A6 implements printable{

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

public static void main(String args[]){

A6 obj = new A6();

obj.print();

}

}

**Java Interface Another Example**

//Interface declaration: by first user

interface Drawable{

void draw();

}

//Implementation: by second user

class Rectangle implements Drawable{

public void draw(){System.out.println("drawing rectangle");}

}

class Circle implements Drawable{

public void draw(){System.out.println("drawing circle");}

}

//Using interface: by third user

class TestInterface1{

public static void main(String args[]){

Drawable d=new Circle();//In real scenario, object is provided by method e.g. getDrawable()

d.draw();

}}