

# Interface

An interface is a kind of class that only contains the methods without any implementation. An interface is a named collection of method definitions (without implementations). An interface can also include constant declarations. Each variable that is defined in interface should be final and static.

An interface is a completely "**abstract class**" that is used to group related methods with empty bodies.

## How to define interface?

```
interface SuperInterface {  
    // final and static variables  
    public void function1();  
    public int function2(int, int);  
}
```

Example:-

```
interface vehicle  
{  
  
    public void dimension(); interface method (does not have a body)  
  
    public void run(); // interface method (does not have a body)  
  
}
```

## Extending interface

An interface can extends another interface but not other classes.

```
interface SubInterface extends SuperInterface{  
    public void function3(int);  
    Public int function4();  
}
```

### **How to implement interface?**

A class never extends an interface rather it implements it. If any class implements any interface than that class should implement each and every function whose definition was in interface. Implementing an interface is like signing a contract with the compiler that states, "I will define all the methods specified by the interface".

```
class A implements SuperInterface{
```

```
public void function1(){  
    // function implementation.  
}  
public int function2(int x, int y){  
    //function implementation.  
    return x + y;  
}  
}
```

Leaving a method of an interface undefined in a class that implements the interface results in a compile error. A class can implement more than one interface by using comma operator.

## Example of Interface

```
// Interface.java
interface PI{
    static final float pi = 3.1416f;
}
interface Area extends PI{
    public float getArea(float x, float y);
}
interface Shape extends Area{
    public void display(float x, float y);
}
class Rectangle implements Shape{
    public float getArea(float x, float y)    {
        return x * y;
    }
    public void display(float x, float y){}
}
class Circle implements Shape{
    public float getArea(float x, float y)    {
        return pi * x * x;
    }
    public void display(float x, float y){}
}
```

```
class Interface {  
    public static void main(String[] args) {  
        Rectangle r = new Rectangle();  
        Circle c = new Circle();  
        Shape s;  
        s = r;  
        System.out.println(s.getArea(2,3));  
        s = c;  
        System.out.println(s.getArea(4,0));  
    }  
}
```

