# **Interface**

Abstract class is used for partial abstraction. Interface is used for full abstraction.

Abstraction is a process where you show only relevant data and hide unnecessary details of an object from the user.

Interface looks like a class but it is not a class. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract (only method signature, no body). Also, the variables declared in an interface are public, static & final by default.

Since methods in interfaces do not have body, they have to be implemented by the class before you can access them. The class that implements interface must implement all the methods of that interface. Also, java programming language does not allow you to extend more than one class, However, you can implement more than one interfaces in your class.

Syntax:

interface DemoInterface{

public void testmethod1();

public void testmethod2();

}

Example:

TestInterface Interface:

**package** FPPackage;

**public** **interface** TestInterface {

**public** **void** test1();

**public** **void** test2();

}

ClassForInterface Class:

**package** FPPackage;

**public** **class** ClassForInterface **implements** TestInterface{

**public** **void** test1() {

System.***out***.println("Implementing test1");

}

**public** **void** test2() {

System.***out***.println("Implementing test2");

}

**public** **static** **void** main(String[] args) {

TestInterface cl = **new** ClassForInterface();

cl.test1();

cl.test2();

}

}

An interface cannot implement another interface. It has to extend another interface.

Example:

Interface TestInterface:

**package** FPPackage;

**public** **interface** TestInterface {

**public** **void** test1();

**public** **void** test2();

}

Interface TestInterface1:

**package** FPPackage;

**public** **interface** TestInterface1 **extends** TestInterface{

**public** **void** test3();

**public** **void** test4();

}

Class ClassForInterface:

**package** FPPackage;

**public** **class** ClassForInterface **implements** TestInterface1{

**public** **void** test1() {

System.***out***.println("Implementing test1");

}

**public** **void** test2() {

System.***out***.println("Implementing test2");

}

**public** **void** test3() {

System.***out***.println("Implementing test3");

}

**public** **void** test4() {

System.***out***.println("Implementing test4");

}

**public** **static** **void** main(String[] args) {

TestInterface cl = **new** ClassForInterface();

cl.test1();

cl.test2();

cl.test3();

cl.test4();

}

}

Even though the class is implementing only TestInterface1 it has to implement all the methods of the TestInterface also as TestInterface1 is extending TestInterface.