# \*\* Interface \*\*

***Interface*** :

1. An **interface** is a blueprint of a class.
2. It has static constants and abstract methods.
3. The interface is a mechanism to achieve [abstraction](https://www.javatpoint.com/abstract-class-in-java).
4. It is used to achieve abstraction and multiple inheritances.

Java Interface also **represents the IS-A relationship**.

Since Java 8, we can have **default and static methods** in an interface.

Since Java 9, we can have **private methods** in an interface.

## Why use interface?

Ans : There are mainly three reasons to use interface.

* Interface is used to achieve abstraction.
* Interface can be used to achieve loose coupling.
* By interface, we can support the functionality of multiple inheritance

## Multiple inheritance is not supported through class in java, but it is possible by an interface, why?

Ans : As we have explained in the inheritance chapter, multiple inheritance is not supported in the case of [class](https://www.javatpoint.com/object-and-class-in-java) because of ambiguity. However, it is supported in case of an interface because there is no ambiguity. It is because its implementation is provided by the implementation class

## What is marker or tagged interface?

Ans : An interface which has no member is known as a marker or tagged interface, for example, [Serializable](https://www.javatpoint.com/serialization-in-java), Cloneable, Remote, etc. They are used to provide some essential information to the JVM so that JVM may perform some useful operation.

//How Serializable interface is written?

**public** **interface** Serializable

{

}

## How to declare an interface?

Ans :

1. To create Interface we have to use Interface Keyword.
2. We can create only abstract Method and variable in Interface, We can’t create Method body in Interface.
3. We can’t instantiated of an Interface just like the abstract class.
4. To access the data of Interface, It is compulsory to implements the interface.
5. To implementation of interface we use interface keyword.
6. It is compulsory to Over Ride all Abstract method which is define in interface.
7. If we can’t over ride all abstract method in (sub-class ko abstract bnana hoga).

**Some Operation On Interface**

1. Class within Interface.
2. Interface within Class.
3. Interface within Interface. (nested interface).
4. Class within class (nested interface).

**Nested Interface in Java**

1. An interface can have another interface which is known as a nested interface.

interface printable

**{**

    void print**();**

    interface MessagePrintable

**{**

        void msg**();**

**}**

**}**

1. The nested interfaces are used to group related interfaces so that they can be easy to maintain.
2. The nested interface must be referred to by the outer interface or class. It can't be accessed directly.

Points to remember for nested interfaces

* The nested interface must be public if it is declared inside the interface,
* but it can have any access modifier if declared within the class.
* Nested interfaces are declared static

Syntax of nested interface which is declared within the class

class class\_name

**{**

**...**

    interface nested\_interface\_name

**{**

**...**

**}**

**}**

interface Showable

**{**

    void show**();**

    interface Message

**{**

        void msg**();**

**}**

**}**

class TestNestedInterface1 implements Showable**.**Message

**{**

    public void msg**(){**System**.**out**.**println**(**"Hello nested interface"**);**

**{**

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

**{**

            Showable**.**Message message**=new**

                        TestNestedInterface1**();**//upcasting here

            message**.**msg**();**

**}**

**}**

**}**

Output : **hello nested interface**

**we can't create method body in interface.**

interface Inter1

**{**

    void show**()**

**{}**

**}**

class Q01\_Interface

**{**

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

**{**

**}**

**}**

error: interface abstract methods cannot have body

        {}

        ^

**we can create abstract method(Without Body) in interface.**

interface Inter1

**{**

    void show**();**

**}**

class Q02\_Interface

**{**

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

**{**

        System**.**out**.**println**(**"Work"**);**

**}**

**}**

Work

**To Access Interface we need to implement the interface, Using implements keywork.**

interface Inter1

**{**

    void show**();**

**}**

class A implements Inter1

**{}**

class Q03\_Interface

**{**

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

**{**

        System**.**out**.**println**(**"Work"**);**

**}**

**}**

error: A is not abstract and does not override abstract method show() in Inter1

class A implements Inter1

^

interface Inter1

**{**

    void show**();**

**}**

class A implements Inter1

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class Q04\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

error: show() in A cannot implement show() in Inter1

    void show()

         ^

interface Inter1

**{**

    void show**();**

**}**

class A implements Inter1

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class Q05\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

Class A

interface Inter1

**{**

    int x **=** 10**;**

**}**

class A implements Inter1

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"X  :  " **+** x**);**

**}**

**}**

class Q06\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

X  :  10

**we can't modify the data of interface,**

**Because interface variable have with 3 keywords**

1. **static,**
2. **final,**
3. **public.**

interface Inter1

**{**

    int x **=** 10**;**

**}**

class A implements Inter1

**{**

    public void show**()**

**{**

        x **=** 20**;**

        System**.**out**.**println**(**"X  :  " **+** x**);**

**}**

**}**

class Q07\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

error: cannot assign a value to final variable x

                x = 20;

                ^

interface Inter1

**{**

    int x **=** 10**;**

**}**

class A implements Inter1

**{**

    public void show**()**

**{**

        int x **=** 20**;**

        System**.**out**.**println**(**"X  :  " **+** x**);**

**}**

**}**

class Q08\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

x = 20;

interface Inter1

**{**

    int x **=** 10**;**

**}**

class A implements Inter1

**{**

    int x **=** 20**;**

    public void show**()**

**{**

        int x **=** 30**;**

        System**.**out**.**println**(**"X  :  " **+** x**);**

        System**.**out**.**println**(**"X  :  " **+** **this.**x**);**

        System**.**out**.**println**(**"X  :  " **+** **super.**x**);**

**}**

**}**

class Q09\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

error: cannot find symbol

    System.out.println("X  :  " + super.x);

                                       ^

interface Inter1

**{**

    int x **=** 10**;**

**}**

class A implements Inter1

**{**

    int x **=** 20**;**

    public void show**()**

**{**

        int x **=** 30**;**

        System**.**out**.**println**(**"X  :  " **+** x**);**

        System**.**out**.**println**(**"X  :  " **+** **this.**x**);**

        System**.**out**.**println**(**"X  :  " **+** Inter1**.**x**);**

**}**

**}**

class Q10\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

x : 30

x : 20

x : 10

interface Inter1

**{**

    int x **=** 100**;**

**}**

interface Inter2

**{**

    int x **=** 200**;**

**}**

class A implements Inter1**,** Inter2

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"X  :  " **+** x**);**

**}**

**}**

class Q11\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

error: reference to x is ambiguous

    System.out.println("X  :  " + x);

                                  ^

---------------------------------------------------------------------------------------------

interface Inter1

**{**

    void show**();**

**}**

interface Inter2

**{**

    void show**();**

**}**

class A implements Inter1**,** Inter2

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"Show Method"**);**

**}**

**}**

class Q12\_Interface

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

Show Method

**Version 1.8 ke Corresponding interface ke ander default method wali bana sakte he.**

interface Inter1

**{**

    default void show**()**

**{**

        System**.**out**.**println**(**"Default Method"**);**

**}**

**}**

class A implements Inter1

**{**

**}**

class Q13\_Interface\_Default\_Method

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

Default Method

---------------------------------------------------------------------------------------------

**Q1. Kya ham Default Method ko Over Ride kar karte he**

**Ans Yes, we can Over Ride Interface Default Method.**

interface Inter1

**{**

    default void show**()**

**{**

        System**.**out**.**println**(**"Default Method Interface Inter1"**);**

**}**

**}**

class A implements Inter1

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"Default Method Class A"**);**

**}**

**}**

class Q14\_Interface\_Default\_Method

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

Default Method Class Ad

**Q2. Why we add default method in interface.**

**Ans Yadi ham chate he implement class me interface me method add karne se koi effect na ho**

**esha liye default method ka use kar sakte he.**

interface Inter1

**{**

    default void show**()**

**{**

        System**.**out**.**println**(**"Default Method Interface Inter1"**);**

**}**

**}**

interface Inter2

**{**

    default void show**()**

**{**

        System**.**out**.**println**(**"Default Method Interface Inter2"**);**

**}**

**}**

class A implements Inter1**,** Inter2

**{}**

class Q15\_Interface\_Default\_Method

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

error: class A inherits unrelated defaults for show() from types Inter1 and Inter2

class A implements Inter1, Inter2

^

interface Inter1

**{**

    default void show**()**

**{**

        System**.**out**.**println**(**"Default Method Interface Inter1"**);**

**}**

**}**

interface Inter2

**{**

    default void show**()**

**{**

        System**.**out**.**println**(**"Default Method Interface Inter2"**);**

**}**

**}**

class A implements Inter1**,** Inter2

**{**

    public void show**()**

**{**

        Inter1**.**super**.**show**();**

        Inter2**.**super**.**show**();**

**}**

**}**

class Q16\_Interface\_Default\_Method

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

Default Method Interface Inter1

Default Method Interface Inter2

**Q3. Kya Static Method ko v Interface ke andar bana sakte he.**

**Ans Yes.**

interface Inter1

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Static Method Interface Inter1"**);**

**}**

**}**

class A implements Inter1

**{**

**}**

class Q17\_Interface\_Default\_Method

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

error: cannot find symbol

                a.show();

                 ^

**Q3. Kya Static Method ko v Interface ke andar bana sakte he.**

**Ans Yes.**

interface Inter1

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Static Method Interface Inter1"**);**

**}**

**}**

class A implements Inter1

**{**

**}**

class Q18\_Interface\_Default\_Method

**{**

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

**{**

        Inter1**.**show**();**

**}**

**}**

Static Method Interface Inter1

interface Inter1

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Static Method Interface Inter1"**);**

**}**

**}**

class A implements Inter1

**{**

    void show**()**

**{**

        Inter1**.**show**();**

        System**.**out**.**println**(**"Class A Show Method"**);**

**}**

**}**

class Q19\_Interface\_Default\_Method

**{**

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

**{**

        A a **=** **new** A**();**

        a**.**show**();**

**}**

**}**

Class A Show Method

**Q4. Kya ham main method ko Interface Bana sakte he**

**Ans Yes**

interface Q20\_Interface\_Main\_Method

**{**

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

**{**

        System**.**out**.**println**(**"Interface Main Method"**);**

**}**

**}**

Interface Main Method

1. Interface ki c ek .class file banti he.

2. Constructor , interface me nai bana sakt ehe kyu ki Constructor ka use instance variable ko initialize ke liye use karte he.

   Or Interface me Instance variable nai hote he,

   (Variable Hote he But bo Static Variable hote he with final Keyword).

**Interface ka referance variable sub-class ke Object ko Hold kar sakta he.**

interface Inter1

**{**

    void show**();**

**}**

class A implements Inter1

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"Show Method"**);**

**}**

**}**

class Q21\_Interface\_Class\_Referance\_Variable

**{**

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

**{**

        Inter1 in **=** **new** A**();**

        in**.**show**();**

**}**

**}**

Show Method

**Class Ke Andar Interface Bana sakte he.**

**( class - > Interface)**

class A

**{**

    interface Inter1

**{**

        void show**();**

**}**

**}**

class B **extends** A

**{**

    public void show**()**

**{**

        System**.**out**.**println**(**"Show Method"**);**

**}**

**}**

class Q22\_Class\_WithIn\_Interface

**{**

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

**{**

        B b **=** **new** B**();**

        b**.**show**();**

**}**

**}**

Show Method

**Interface Ke Andar Interface Bana sakte he. ( interfaec- > Interface )**

interface A

**{**

    void show1**();**

    interface Inter1

**{**

        void show2**();**

**}**

**}**

class B1 implements A

**{**

    public void show1**()**

**{**

        System**.**out**.**println**(**"Show Method - 1"**);**

**}**

**}**

class B2 implements A**.**Inter1

**{**

    public void show2**()**

**{**

        System**.**out**.**println**(**"Show Method - 2"**);**

**}**

**}**

class Q23\_Interface\_WithIn\_Interface

**{**

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

**{**

        B1 b1 **=** **new** B1**();**

        b1**.**show1**();**

        B2 b2 **=** **new** B2**();**

        b2**.**show2**();**

**}**

**}**

Show Method - 1

Show Method – 2

**Interface Ke Andar Class Bana sakte he.**

**( Interface -> class )**

interface Inter1

**{**

    class A

**{**

        void show1**()**

**{**

            System**.**out**.**println**(**"Show Method - 1"**);**

**}**

**}**

**}**

class Q24\_Interface\_WithIn\_Class

**{**

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

**{**

        Inter1**.**A a **=** **new** Inter1**.**A**();**

        a**.**show1**();**

**}**

**}**

Show Method - 1

**Interface Ke Andar Class Bana sakte he.**

interface Inter1

**{**

    class A

**{**

        void show1**()**

**{**

            System**.**out**.**println**(**"Show Method - 1"**);**

**}**

**}**

**}**

class B implements Inter1

**{}**

class Q25\_Interface\_WithIn\_Class

**{**

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

**{**

        // 01

        // B b1 = new B();

        // b1.show1();

        // 02

        // A.B b2 = new A.B();

        // b2.show1();

        // 03

        B**.**A b3 **=** **new** B**.**A**();**

        b3**.**show1**();**

**}**

**}**

// 01

 error: cannot find symbol

    b1.show();

    ^

// 02

error: cannot find symbol

    A.B b2 = new A.B();

     ^

// 03

Show - 1

**Yah Class A ko inherite kiy he na ki interface ko.**

interface Inter1

**{**

    class A

**{**

        void show1**()**

**{**

            System**.**out**.**println**(**"Show Method - 1"**);**

**}**

**}**

**}**

class B **extends** Inter1**.**A

**{}**

class Q26\_Interface\_WithIn\_Class

**{**

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

**{**

        // 01

        B b1 **=** **new** B**();**

        b1**.**show1**();**

**}**

**}**

// 01

Show - 1