**Abstraction**

* >Hiding internal knowledge and showing external things or functionality only
* > abstraction can be done in two ways

1.Abstract class

2.interface

### **Abstract class**

* A class which is declared as abstract is known as an **abstract class**. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be
* An abstract class must be declared with an abstract keyword.
* It can have abstract and non-abstract methods.
* It cannot be instantiated.
* It can have [constructors](https://www.javatpoint.com/java-constructor) and static methods also.
* It can have final methods which will force the subclass not to change the body of the method.

**Syntax:**

Public abstract class ClassName //abstract class declaration

{

Public abstract void display(); //abstract method declaration

**Abstract class has two methods**

1. Abstract method(with out body)
2. 2.non abstract method(with body)

/\*Abstraction

\* declared with keyword abstract.

\* contain a abstract and non abstract method

\* static method has no body and is end with ;

\*/

**Class 1**

package abstraction;

public abstract class Absta {

public abstract void print(); //abstract method,has no body and end with ;

public void display()

{

System.out.println("this is a non abstract method");

}

}

**Class 2**

package abstraction;

public class Absta1 extends Absta {

public void print()

{

System.out.println("this is a abstract method");

}

public static void main(String[] args) {

Absta1 obj=new Absta1();

obj.display();

obj.print();

}}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2-2-2022\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Interface**

* It is a blueprint of a class.
* It has static constants and abstract methods.
* The interface in Java is *a mechanism to achieve* [*abstraction*](https://www.javatpoint.com/abstract-class-in-java).
* There can be only abstract methods in the Java interface, not method body.
* It is used to achieve abstraction and multiple inheritance
* Java Interface also **represents the IS-A relationship**.

## **Use of Java 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.

## **Declaration of an interface**

Interface <interface name>

**Method declaration**

**Syntax :** Accessmodifier abstract retuntype methodname ();

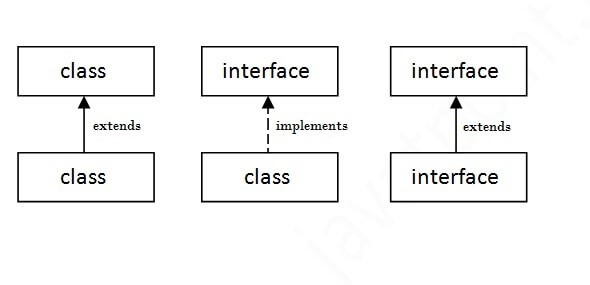
**Example :** public abstract void print();

**Variable declaration**

**Syntax :** Accessmodifier static final datatype variablename ();

**Example :** public static final int a=20;

#### ***Relationship between classes and interfaces***



**Differences between abstract class and interface**

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also. |
| 2) it **doesn't support multiple inheritance**. | it **supports multiple inheritance**. |
| 3) it **can have final, non-final, static and non-static variables**. | It has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword "implements". |

//prgrm

//class 1

package interfacepackage;

public interface InterfSample {

public static final int a=10;//interface variable

int b=20;//instance variable

public abstract void display(); //abstract method

public void add();//abstract method

}

//class 2

package interfacepackage;

public class Interfaclass implements InterfSample {

public void display()

{

System.out.println("abstract method");

}

public void add()

{

int x=10,y=20;

int z=a+b;

System.out.println("Sum : "+z);

}

public void childmethod()

{

int c=b-a;

System.out.println("Diff : "+c);

System.out.println("child method");

}

public static void main(String[] args) {

Interfaclass obj=new Interfaclass();

obj.display();

obj.add();

obj.childmethod();

System.out.println("a : "+a);

System.out.println("b : "+obj.b);

}

}