* Interface :-

An interface is a collection of abstract methods which are public in scope.

(or)

It is a collection of methods, which are public and abstract by default.

Abstract method :-

It is a method with no body (or) implemenation, ie there is no code at all associated with method.

↳ The best part of an interface is that a class can inherit any number of interfaces, thus allowing multiple inheritance in java.

↳ Java does not support multiple inheritance among classes, but interfaces allow java to support this feature.

→ Interfaces are declared with help of keyword "Interface".

Note :- In interface, None of methods have body.

Syntax :-

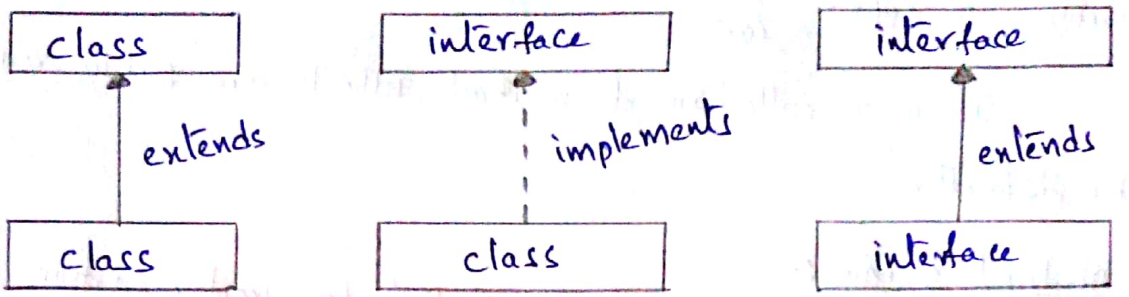
```
interface interfacename
{
returntype methodname(arguments);
=
=
}
```

* In simple words, Interface is a blueprint of class, it has static constants and abstract methods.

* The interface in java is a mechanism,
  - ↳ to achieve abstraction
  - ↳ to achieve multiple inheritance

Note :- It is mandatory to add the access specifier public to the method declaration, otherwise the compiler will not compile the program (The older version of java, i.e before java 8).

* In Java, a class extends another class, an interface extends another interface but a class implements an interface.



Example:-

Calculator.java

```
interface CalInterface
{
 int add (int a, int b);
 int sub (int a, int b);
}
class Calculator implements CalInterface
{
 public int add (int a, int b)
 {
  return a+b;
 }
 public int sub (int a, int b)
 {
  return a-b;
 }
 public static void main(String args [])
 {
  Calculator cal = new Calculator();
  System.out.println ("value after addition =" + cal.add (5,2));
  System.out.println ("value after Substraction =" + cal.sub(5,2));
 }
}
```

output:- javac Calculator.java
java Calculator
value after addition = 7
value after Substraction = 3

* Extending interfaces :- (or) Inheritance in interfaces

Just like inheritance in classes, the interfaces can also be extended. An interface can inherit another interface using the same keyword "extends".

Example :-

InherDemo.java

```
interface A
{
  void showA();
}
interface B extends A
{
  void showB();
}
class InherDemo implements B
{
 public void showA()
 {
  System.out.println("Method of interface A");
 }
 public void showB()
 {
  System.out.println("Method of interface B");
 }
public static void main(String args[])
{
 InherDemo d = new InherDemo();
  d.showA();
  d.showB();
}
}
```

output:- javac InherDemo.java

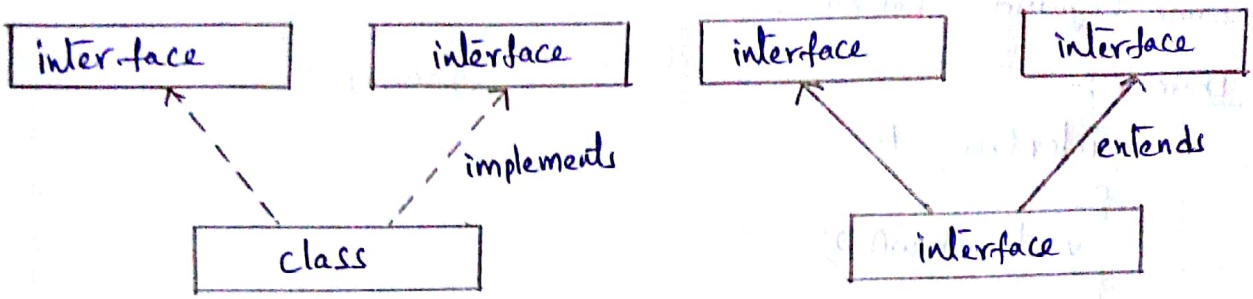
java InherDemo

Method of interface A

Method of interface B

* Multiple inheritance in java by interface :-

If a class implements multiple interfaces (or) an interface extends multiple interfaces is known as Multiple inheritance.



Example :- MultipleInheritanceDemo.java

```
interface printable
{
  void print();
}
interface Showable
{
  void show();
}
class MultipleInheritanceDemo implements printable, Showable
{
public void print()
{
 System.out.println("Hai");
}
public void show()
{
 System.out.println("Welcome");
}
public static void main(String args[])
{
MultipleInheritanceDemo obj = new MultipleInheritanceDemo();
  obj.print();
  obj.show();
}
}
```

output :- javac MultipleInheritanceDemo.java
java MultipleInheritance
Hai
Welcome.