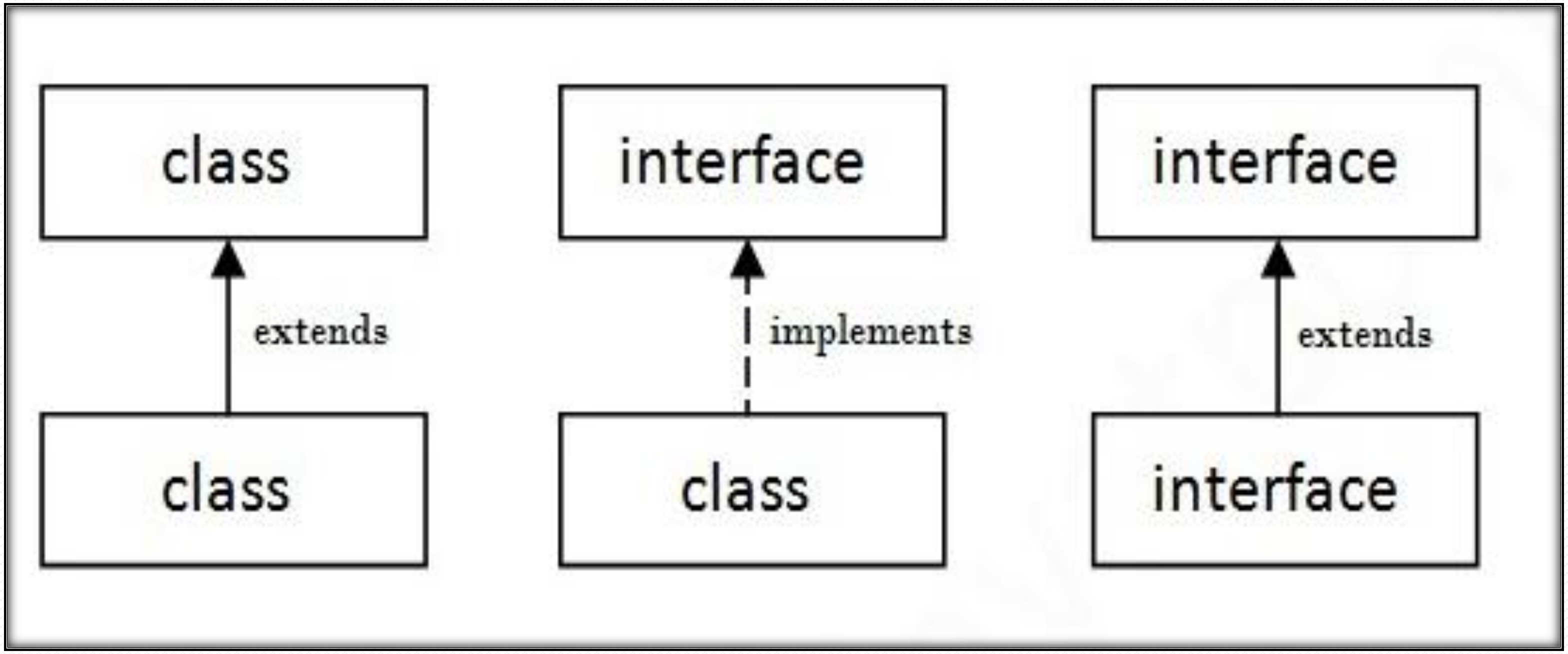
Understanding Interfaces

An interface in the Java programming language is an abstract type that is used to specify a behavior that classes must implement.

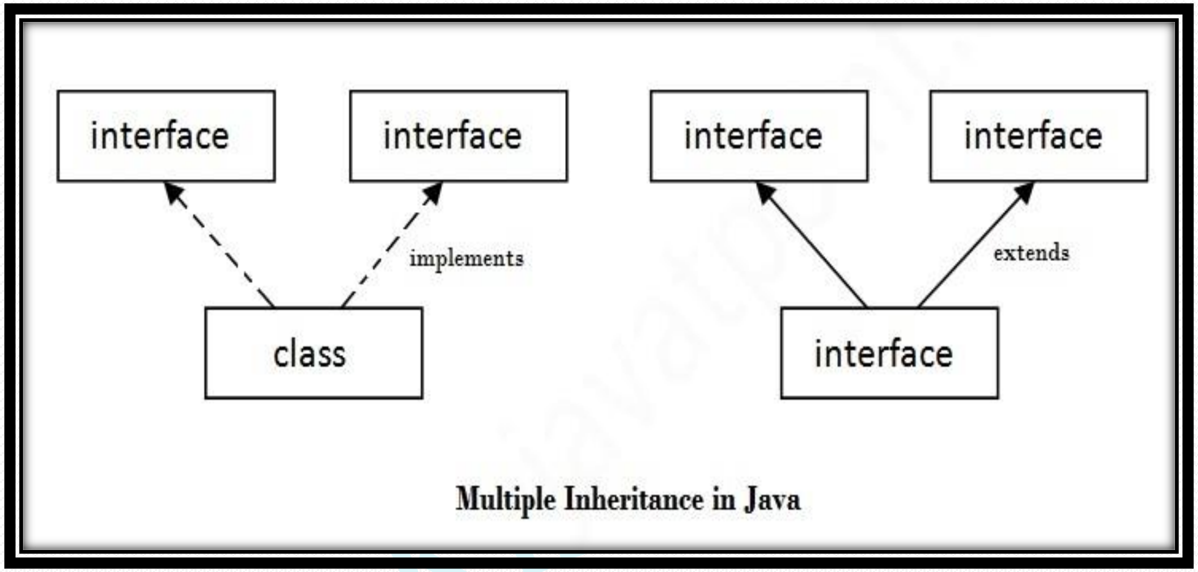
class	interface
We can instantiate a class, i.e. we can create an object reference for a class	We can't instantiate an interface.
A class is declared using a keyword '**class**' class <class name> { }	An interface is declared by using a keyword '**interface**'. interface <interface name>{ }
The members of a **class** can have the access modifiers like **public, private, protected.**	The members of an **interface** are always **public**. **Up to (1.7v)**
Inside a class you can have method implementation.	In interfaces we can't write method body, because all the methods are by default public abstract methods. **Up to (1.7v)**
Multiple inheritance is not possible	Multiple inheritance is possible
We can have constructor.	We can't have constructors

- An interface is a blueprint of a class.
- Interface is a mechanism to achieve fully abstraction in java.
- There can be only abstract methods in the interface (up-to 1.7v).
- It is used to achieve fully abstraction and multiple inheritance in Java.
- Interface fields are public, static and final by default, and methods are public and abstract.
- For variables present in the interface we should provide initialization at the time of declaration only.
- Interfaces should have only abstract methods. (we can check the internal implementation by using javap .className).
- We can declare a class inside the interface.
- We can write main method in interface from **jdk 1.8v**

- If we want to inherit an interface from a class we need to use the keyword ‘implements’ not ‘extends’.
- If we want to inherit an interface from another interface we need to use the keyword ‘extends’ not ‘implements’.



- We can achieve multiple inheritance in java by using interfaces.



Multiple Inheritance in Java

- We can extends multiple classes from a class at a time (T/F)
- We can implement any number of interfaces from a interface at a time (T/F)
- We can implement only one interface from a class at a time (T/F)

- When implementing an interface , if a class cant provide the implementation of all the abstracts methods present in that interface then make that class as abstract class.

Marker Interface:

- A marker interface in Java is an interface with no fields or methods. To be precise, an empty interface in Java is called a marker interface.
- Examples of marker interfaces are Serializable, Cloneable etc