**Interface in Java8**

**Until 1.7, Interface can only contain abstract method declaration.**

In java-1.8 has concept of Stream API. Stream API contains many methods that works with collection.

For Example if list earlier had x methods. Then after introduction of stream list now has ‘x+y’ methods.

Now, in order to continue providing backward support Oracle, instead of declaring abstract methods inside interface, so that the backward classes that implemented list interface need not to define those methods inside them. Oracle has defined these methods in Interface only.

**Example –**

**Interface** <interface-name> {

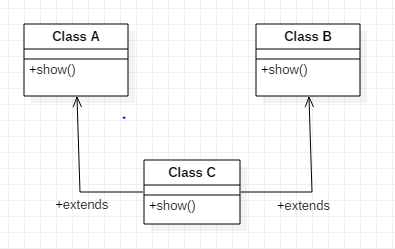
public void add ();

default void show () {

// whenever define method in interface make sure to include default keyword..

}

}

**Diamond Problem –**

IF Class C extends Class A and B both then in class C if show() is called then which show will be called of class A or class B.

Hence Multiple Inheritance is not applicable in java.

But Multiple inheritance can be implemented using Interface in java 8.

Hence,

Let’s interface-A and interface-B and Class C implements both these interfaces.

Now –

If Interface-A has Show() method but not interface-B then no confusion and Class C can implement both these interface.

But, if interface-B also has show() method then that show() method is needed to be defined in class C even if it’s definition is already present in any of the or both interfaces.

So if class c object call show method then class c defined show method will be called.

**Next Scenario-**

If-

Interface-A has show() defined;

Class –A has show() defined

Class c extends A and Implements A.

And Class C object calls show();

Then which show will be called Class A or Interface show(); ?

**Ans** –

Class –A because class has more power than interface.

Class has high priority than interface and Class A method will hide interface method if of same name.

**Next Scenario-**

If your interface has any method that is already defined / declared in Object class then it will not work.

**Example –**

Interface A{

defaut boolean equal(Object o){

return true;

}

}

This will not work as equal is already defined in Object Class. Hence will give error.

**Next Scenario-**

Static methods in JAVA 8

Interface I{

default static void show(){

// definition

}

}

To call it use I.show();