Java 8 Features

Interface : Till Java 7 interface is known as 100% pure abstract class. Which contains only abstract method. From Java8 onward interface can contains method with body. But the method is must with default or static keyword.

Interface contains default method with default implementation then the class which implements that interface no need to provide the body. If we need to override we can override default method also.

Static method we can’t override. If interface contains method with body with keyword as static we can’t call that interface through object of that class. The class which implements that interface. Static method which belong to interface we have to class through interface name.

Bank interface

withdraw

deposit

default gst 12%

static home loan : 8%

HDFC implements Bank

Provided body for withdraw and deposit

HSBC implements Bank

Provided body for withdraw and deposit

SBI implements Bank

Provided body for withdraw and deposit

Functional interface.

The interface contains only one abstract method is known as functional interface. It can contains more than one default as well as static method but only one abstract method.

Marker interface The interface contains zero method or no method is known as marker interface. Example : Clonenable or Serializable.

Java 8 provided pre defined annotation @FuntionalInterface. This annotation we can use on interface level to check is that interface is functional interface or not.

Inner class :

Class within another class is known as inner class.

1. Non static inner class
2. Static inner class
3. Anonymous inner class
4. Local method class

Lambda Expression : Lambda is a Greek word. Using lambda expression Java also become functional programming language.

Function within another function or method within another method.

void display() {

void display2() {

}

}

From Java8 onward using lambda expression we can achieve functional programming concept. Lambda expression is anonymous function or method.

If we can lambda expression for those method. The method must be part of interface and that interface must be functional interface.