**Java 8**

1. From java 8 onward interface can contains method with body. But method must be default or static.

Default method we can use it as well as we can override.

But static method we can’t override.

Static method we need to call through interfaceName.methodName().

Default method we can call with help of object.

interface Bank {

public void withdraw();

public void deposit();

public default void findBalance(int accno){

provide default implementation.

}

public static void roi() {

static implementation.

}

}

class Hsbc implements Bank {

public void withdraw(){}

public void deposit(){}

}

Class BankOfAmerica implements Bank {

public void withdraw(){}

public void deposit(){}

}

1. Functional interface : the interface contains only one abstract method. it can contains more than one default as well as static but only one abstract method that type of interface is known as functional interface.

Java 8 provide new annotation @FuntionalInterface. This annotation we can write on interface. If interface is a type of functional interface we doesn’t get any error else we get error.

1. Lambda expression : using Lambda expression we are provide the body for method without name ie anonymous method. Lambda expression is like a arrow function in JavaScript or TypeScript or angular. If we want to use lambda expression we need functional interface.

Lambda expression return the value by default without return keyword. if we want to write only 1 line statement then curly braces not required and return keyword not required. If we want to write more than one statement then you can use curly braces and you need to return output with return keyword explicitly.

Syntax

InterfaceName referenceName = (parameter)->body;