Java 8 and 11

From java8 onward

1. From java8 onward interface can contains method with body. But we need to use static or default keyword.

```
interface Bank {
    void withdraw();
    void deposit();
    default void rateOfInterest(){}
}
class Hsbc implements Bank {
            void withdraw(){}
            void deposit(){}
}
class Sbi implements Bank {
            void withdraw(){}
            void deposit(){}
}
```

Default method we can override but static method we can't override part of java8 interface.

2. 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.
3. Lambda expression

   But from java8 onward java also known a functional programming language(we can use nested function features).

   If we want to achieve lambda expression we need functional interface.
   Lambda is like an arrow function in JavaScript.

   Lambda expression also known as anonymous method or functions.
   Syntax

   InterfaceName refereceName = (parameterList)->body;

   Lambda expression method return data without return keyword like arrow.