A lambda expression can take any number of parameters (see below 1 or 2.. any)

v1 = () -> System.***out***.println("Hello World!");

v2 = (s) -> s.length();

v3 = (i, j) -> i + j;

If multiple parameters are present then the parameters should be separated with, (I,j)

If only one parameter is there, then parenthesis is optional

Usually we can specify the type of the parameter, if compiler expects the type of the context, then we can remove type [Type Inference]

(**int** i, **int** j) -> i + j; or (i, j) -> i + j;

If lambda expression body contains more than on statement then we need curly braces. If we have only one line then its optional

(i, j) -> i + j;

Or

(i, j) -> {

i + j;

System.***out***.println("Hello World!");

};

If lambda expression returns something, then we can remove return statement

(s) -> s.length();

===

Functional interface

we need function interface to invoke lambda expression.

An interface with single abstract method is called Functional interface (see below)

Note: A functional interface may contain more than one static method or default method, but should contain only one abstract method.

Below are some functional interfaces that already exist in Java

**public** **interface** ActionListener **extends** EventListener {

**public** **void** actionPerformed(ActionEvent e);

}

**public** **interface** Runnable {

**public** **abstract** **void** run();

}

**public** **interface** Runnable {

**public** **abstract** **void** run();

}

**public** **interface** Comparable<T> {

**public** **int** compareTo(T o);

}

Note: A functional interface may contain more than one static method or default method, but should contain only one abstract method.

**interface** MyFunctionalInterface1 {

**public** **void** printIt();

**public** **static** **void** test() {

System.***out***.println("Hello World!");

}

**public** **static** **void** test2() {

System.***out***.println("Hello World!");

}

**public** **default** **void** test3() {

System.***out***.println("Hello World!");

}

**public** **default** **void** test4() {

System.***out***.println("Hello World!");

}

}