**Java-8 – Functional Interface**

* **Characteristics of Lambda expression:**

1. Lambda expressions can take any number of parameters.

() -> System.out.println(“hello”);

(a, b) -> System.out.println(a+b);

s -> s.length();

1. If multiple parameters present then those parameters should be separated with ,

Example: (a,b) -> System.out.println(a+b);

1. If only one parameter available then parenthesis is optional

(s) -> s.length() 🡺 s -> s.length

1. Usually we can specify the type of parameter. The compiler can guess the parameter type based on context then we can remove the type. [Type Inference]

(int a, int b) -> System.out.println(a+b)

(a, b) -> System.out.println(a+b);

1. Similar to method body, lambda expression body can contain any number of statements. If multiple statements are there then we should enclose with curly braces.

() -> {

Statement1;

Statement2;

Statement3;

}

If body contains only one statement then curly braces are optional.

() -> System.out.println(“Hello”);

1. If lambda expression returns something, then we can remove return keyword.

s -> s.length();

* **Functional Interface:**

An interface which contains Single Abstract Method

We can use functional interface to invoke lambda expression.

Example:

Runnable

Callable

ActionListener

Comparable