



<http://www.cinstitute.co.in>
<https://youtube.com/tarunsir>

+917742817777

Lambda in java

Hello guys, in this video you will learn about Lambda and Functional interfaces.

A functional interface is an interface that has only a single abstract method declaration. The functional interface can have any number of default methods and static methods.

An interface that has a single abstract method was commonly called a **Single Abstract Methods** or **SAM** type in java7 and prior.

Java8 introduced the `@FunctionalInterface` annotation which forces the interface to have a single abstract method only otherwise it produces a compilation error.

Lambda Expression

Lambda expression is, essentially, an anonymous or unnamed method. The lambda expression does not execute on its own. Instead, it is used to implement a method defined by a functional interface.

Lambda allows us to treat functionality as the function argument or the code as the function argument. It is a function that belongs to no class.

Creating lambda expressions

The lambda expression has the following syntax:

(parameters) -> {lambda-code}

You can omit the curly braces if there is only a single statement in the lambda-code.

Let's understand with a full example:

```
interface I1
```



<http://www.cinstitute.co.in>
<https://youtube.com/tarunsir>

+917742817777

```
{  
    public void display();  
}
```

class LambdaDemo

```
{  
    public static void main(String args[])  
    {  
        I1 obj1=()->System.out.println("Hello this is a simple lambda");  
        obj1.display();  
        I1 obj2=()->System.out.println("Bye this is a simple labda");  
        obj2.display();  
    }  
}
```

Example2:

```
interface I1  
{
```



<http://www.cinstitute.co.in>
<https://youtube.com/tarunsir>

+917742817777

```
public int calculate(int a,int b);
```

```
}
```

```
class LambdaDemo
```

```
{
```

```
    public static void main(String args[])
```

```
    {        int x=100,y=20;
```

```
        l1 add=(a,b)->{return a+b;;};
```

```
        l1 sub=(a,b)->{return a-b;;};
```

```
        l1 mul=(a,b)->{return a*b;;};
```

```
        l1 div=(a,b)->{return a/b;;};
```

```
        System.out.println(add.calculate(x,y));
```

```
        System.out.println(sub.calculate(x,y));
```

```
        System.out.println(mul.calculate(x,y));
```

```
        System.out.println(div.calculate(x,y));
```

```
}
```



<http://www.cinstitute.co.in>
<https://youtube.com/tarunsir>

+917742817777

}

Java By Tarun Sir