Functional Interface without Arguments

In previous posts we saw what [Lambda Calculus](http://data-structure-learning.blogspot.com/2015/05/lambda-calculas.html) is, [Imperative and Declarative Programming](http://data-structure-learning.blogspot.com/2015/06/java-imperative-and-declarative.html), [Functional Interfaces](http://data-structure-learning.blogspot.com/2015/06/functional-interfaces-java-8.html) & [using Functional interface](http://data-structure-learning.blogspot.com/2015/06/using-functional-interface.html).

Let us now dive into developing our own Functional Interface.

We will design an interface and make sure it is Functional Interface. Then we will use Lambda Operator to call the method of the interface.

Below is the Functional Interface. We also use annotation for it.

**package** com.example.javase8.interfaces;

@FunctionalInterface

**public** **interface** SimpleInterface {

**public** **void** doSomething();

}

Now we will write a class the uses this interface.

**package** com.example.javase8;

**import** com.example.javase8.interfaces.SimpleInterface;

**public** **class** UseSimpleInterface {

**public** **static** **void** main(String[] args) {

SimpleInterface obj = () -> {

System.***out***.println("Inside Lambda");

};

obj.doSomething();

}

}

It outputs

Inside Lambda

In next post we will see how to build a Functional Interface with arguments.