SHAIK md sameer

[Email address]

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

*Lambda Expressions in Java*

# 1. Lambda Expressions in Java

1. As of our knowledge we know Java is an Object-Oriented Programming Language
2. In Java – Oops Concept – Data is saved through Class and Objects and processing also happen through Objects
3. By default, java Does not Support Functional Programming

## 1.1 Why Lambda Expressions introduced in Java

1. *To bring the Functional Programming feature in Java – Lambda Expressions are introduced in Java.*
2. *Other main Objective is Code optimization.*
3. In Functional Programming Data is Stored in the form of Functions and Variables
4. *Lambda Expression is an Anonymous function or method*
5. *In our General Language we call Methods in Java, but here in Lambda we call Methods as function*

## 1.2 Anonymous Function

1. What is Anonymous function means – Name Less function
2. Function Does not have Return type and Access Modifier
3. Function does not return any Value

## 1.3 Functional interface (follow Single- Abstract-Method)

1. Before java version 7 – In Interface we can store only abstract methods
2. But after version 7- In Interface we can store Default and Static methods
3. Functional interface means – Any interface contain only one Abstract method and N number of default and static methods is called as Functional interface.
4. We have so many Functional interfaces in Java

Example Runnable interface

Functional interface ------Abstract Method (single)

1. Runnable interface – is a default functional interface – Abstract method – run ()
2. Callable interface –call ()
3. Comparable – compared to ()
4. Action listener - action performed ()

#### 1.3.1Why functional interface

1. To call Lambda Expression we use Functional interface only
2. With-out Functional interface we cannot invoke the lambda Expression

#### 1.3.2Predefined functional interface

|  |  |
| --- | --- |
| S.no | Pre-defined Functional interfaces Package - java.Util Function |
| 1 | Predicate |
| 2 | Function |
| 3 | Consumer |
| 4 | Supplier |

## 2. Predicate

1. Predicate is a default functional interface
2. Having single abstract method
3. Default method in Predicate is test ()
4. Predicate will return a Boolean value – True or false based on condition

# 