

Lambda Functions in Java

Student's Name

Department, Institutional Affiliation

Course Number and Name

Instructor's Name

Due Date

Lambda Functions in Java

A lambda function in java is a code that returns a value by taking in parameters. The difference between a lambda expression and a method is that a lambda expression does not need a name and can be declared inside a method. If a lambda expression has more than one parameter, the parameters are wrapped using parentheses (Kanjilal, 2022). As a result, lambda expressions do not contain statements and loops such as “if” or “for”, respectively.

There are fewer lines of code used in Lambda expressions. According to Oral 2021, lambda expressions reduce the amount of code required. For instance, a single line of code is used in a lambda expression to print all the elements of an array. Java programs that utilize lambda expressions run faster than those that do not.

An arrow is used to divide the lambda expression into the parameters section and the body. It can have no parameters, one parameter, or multiple parameters. The body of a lambda expression is surrounded by curly braces (Oral, 2021). As a result, it is easy to implement lambda expressions inside methods in Java.

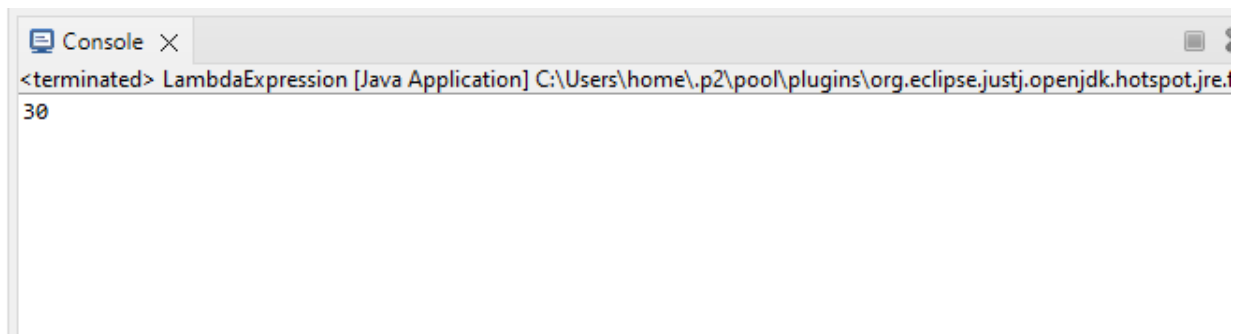
Demonstration of Lambda Expressions in Java

Lambda Expression Using an Interface and Two Arguments

```
1 interface Add{
2     int add(int a,int b);
3 }
4
5 public class LambdaExpression {
6     public static void main (String[] args)
7     {
8         Add add1 =(a,b)->(a+b);
9         System.out.println(add1.add(10,20));
10    }
11
12
13
14 }
15
```

Figure 1 Java Code Demonstrating the Use of a Lambda Expression with an Interface

Output While the Code is in Run



```
<terminated> LambdaExpression [Java Application] C:\Users\home\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.
30
```

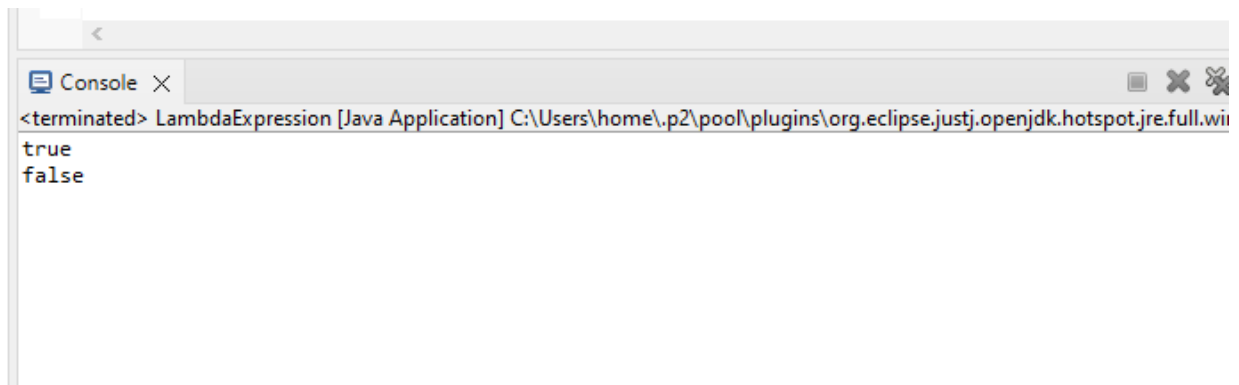
Figure 2 The Output

Lambda Expression without Arguments

```
1 import java.util.function.BooleanSupplier;
2
3 public class LambdaExpression {
4     public static void main (String[] args)
5     {
6         BooleanSupplier bs = () -> true;
7         System.out.println(bs.getAsBoolean());
8
9         int x = 0, y= 1;
10        bs = () -> x > y;
11        System.out.println(bs.getAsBoolean());
12    }
13
14
15
16 }
17
```

Figure 3 Lambda Express with No Parameter

Output While the Code is in Run



```
<terminated> LambdaExpression [Java Application] C:\Users\home\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi
true
false
```

Figure 4 The Output while the Lambda Expression with No Parameter is Run

References

Kanjilal, J. (2020, May 20). Arrays in Java: Declare, Define, and Access Array. Simplilearn.com.

<https://www.simplilearn.com/tutorials/java-tutorial/arrays-in-java>

Oral, M. E. (2021, March 28). Java Lambda Expressions. Medium.

<https://medium.com/javarevisited/java-lambda-expressions-6696793a6c38>