

Java-Success.com

Industrial strength Java/JEE Career Companion for those who want to go places

[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

[Home](#) > [Interview](#) > [Core Java Interview Q&A](#) > [FP](#) > Java 8: What is currying?

Does Java 8 support currying?

Java 8: What is currying? Does Java 8 support currying?

Posted on [November 8, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — No

[Comments](#) ↓

Currying (named after Haskell Curry) is the fact of evaluating function arguments one by one, producing a new function with one argument less on each step.

Java 8 still does not have first class functions, but currying is “practically” possible with verbose type signatures, which is less than ideal. Here is a very simple example:

```
1 package com.java8.examples;
2
3 import java.util.function.Function;
4
5 public class CurryTest {
6
7     public static void main(String[] args) {
```

[9 tips to earn more](#) | [What can u do to go places?](#) | **945+** members. [LinkedIn Group](#). [Reviews](#)

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

[+ Ice Breaker Interview](#)

[+ Core Java Interview C](#)

[+ Java Overview \(4\)](#)

[+ Data types \(6\)](#)

[+ constructors-methc](#)

[+ Reserved Key Wor](#)

[+ Classes \(3\)](#)

[+ Objects \(8\)](#)

[+ OOP \(10\)](#)

[+ GC \(2\)](#)

[+ Generics \(5\)](#)

```

8     Function<Integer,Function<Integer,Integer>> add = addOne.apply(1);
9
10    Function<Integer,Integer> addOne = add.apply(1);
11    Function<Integer,Integer> addFive = add.apply(1);
12    Function<Integer,Integer> addTen = add.apply(1);
13
14    Integer result1 = addOne.apply(2); // returns 3
15    Integer result2 = addFive.apply(2); // returns 5
16    Integer result3 = addTen.apply(2); // returns 12
17
18    System.out.println("result1 = " + result1);
19    System.out.println("result2 = " + result2);
20    System.out.println("result3 = " + result3);
21
22 }
23 }
24

```

The output will be:

```

result1 = 3
result2 = 5
result3 = 12

```

```

1 System.out.println(add.apply(1).apply(3)); // returns 12
2

```

If you want to add 3 numbers

```

1 Function<Integer,Function<Integer,Function<Integer,Integer>>> add = addOne.apply(1).apply(3);
2
3 System.out.println(add.apply(1).apply(3).apply(5));
4

```

Note that the **Function** interface is in Java 8 API.

Now Currying in **Pre Java 8** with anonymous inner classes

Define your own **Function** interface.

```

1 package com.currying;
2
3 public interface Function<A,B> {
4     B apply (A a);
5 }

```

FP (8)

- 01: ♦ 19 Java 8 Interview Questions
- 02: ♦ Java 8 Stream API
- 03: ♦ Functional Programming
- 04: ♥♦ Top 6 tips for Java 8
- 05: ♥ 7 Java FP Interview Questions
- Fibonacci number using Java 8
- Java 8 String stream
- Java 8: What is currying?

IO (7)

- Multithreading (12)
- Algorithms (5)
- Annotations (2)
- Collection and Data Structures
- Differences Between Java 7 and 8
- Event Driven Programming
- Exceptions (2)
- Java 7 (2)
- Java 8 (24)
- JVM (6)
- Reactive Programming
- Swing & AWT (2)
- JEE Interview Q&A (3)
- Pressed for time? Java Interview Questions
- SQL, XML, UML, JSC
- Hadoop & BigData Interview Questions
- Java Architecture Interview Questions
- Scala Interview Q&As
- Spring, Hibernate, & JPA
- Testing & Profiling/Selenium
- Other Interview Q&A (1)
- Free Java Interview Questions

As a Java Architect

[Java architecture & design concepts](#)
[interview Q&As with diagrams](#) | [What should](#)

6
7[be a typical Java EE architecture?](#)

Provide the implementation class for Function with **anonymous inner** classes

```

1 package com.currying;
2
3 public class FunctionImpl implements Function<Integer, Integer> {
4
5     @Override
6     public Integer apply(Integer a) {
7         return a;
8     }
9
10    public static Function<Integer, Integer> add() {
11        return new Function<Integer, Integer>() {
12
13            @Override
14            public Integer apply(Integer x) {
15                return new Function<Integer, Integer>() {
16
17                    @Override
18                    public Integer apply(Integer y) {
19                        return x + y;
20                    }
21                };
22            }
23        };
24    }
25
26    public static void main(String[] args) {
27        System.out.println(add());
28        System.out.println(add().apply(1));
29        System.out.println(add().apply(1).apply(2));
30        System.out.println(add().apply(0).apply(2));
31    }
32 }

```

Senior Java developers must have a good handle on

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3\)](#)
- [Java Debugging \(21\)](#)
- [Judging Experience \(1\)](#)
- [Low Latency \(7\)](#)
- [Memory Management \(1\)](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Management \(1\)](#)

Output is:

```

com.currying.FunctionImpl$1@73d16e93
com.currying.FunctionImpl$1$1@659e0bfd
3
2

```

Tedious hey!!! compared to Java 8. It is even much easier in pure functional languages.

80+ step by step Java Tutorials

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2\)](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tutorials \(1\)](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)

Popular Posts

♦ 11 Spring boot interview questions & answers

861 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

829 views

18 Java scenarios based interview Questions and Answers

448 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

407 views

♦ 7 Java debugging interview questions & answers

311 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

303 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

294 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

288 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

263 views

8 Git Source control system interview questions & answers

215 views

✚ Spring & Hibernate Ti

✚ Tools Tutorials (19)

✚ Other Tutorials (45)

Preparing for Java written & coding tests

open all | close all

✚ ♦ Complete the given

✚ Can you write code? (

✚ Converting from A to I

✚ Designing your classe

✚ Java Data Structures

✚ Passing the unit tests

✚ What is wrong with th

✚ Writing Code Home A

✚ Written Test Core Jav

✚ Written Test JEE (1)

How good are your...to go places?

open all | close all

✚ Career Making Know-

✚ Job Hunting & Resum

Bio

Latest Posts



Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in



2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.

**About** [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](#) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.

◀ Java 8: 7 useful miscellaneous additions you must know

Java 8: Does "Optional" class alleviate the pain of ubiquitous

NullPointerException? ▶

Posted in FP, Java 8, member-paid

Leave a Reply

Logged in as [geethika](#). [Log out?](#)

Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)
☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.