

Industrial strength Java/JEE Career Companion to open more doors

[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](#) › [Java 8](#) › Top 8 Java 8 feature list

Top 8 Java 8 feature list

Posted on [November 25, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — No

[Comments](#) ↓

0

Like

Share

Tweet

0

G+1

Share

Are you ready to get cracking with Java 8 new features? Here are a comprehensive Java 8 features, tutorials, and examples. Drill down into each.

#1: Transforming your thinking from OOP to FOP: One needs to get used to the transformation from imperative programming to functional programming. You like it or not, you will be using functional programming in **Java 8**, and interviewers are going to quiz you on functional programming. Fortunately, Java is not a fully functional programming language, and hence one does not require the full leap to functional programming. Java 8 supports both imperative and functional programming approaches.

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 \(5\)](#)

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

[+ IO \(7\)](#)

[+ Multithreading \(12\)](#)

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

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

[+ Collection and Dat](#)

[+ Differences Between](#)

[+ Event Driven Progr](#)

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

[+ Java 7 \(2\)](#)

#2: Here are top 6 Java 8 features you can start using now by downloading Java 8.

1. Interface can have static and default methods.
2. Lambda expression for supporting closures.
3. Java 8 adopts Joda.
4. Java 8 corrects its lack of good support for JavaScript with the engine **Nashorn**.
5. Parallel processing piggy backing the [Java 7 Fork/Join](#).
6. The `java.util.concurrent.atomic` package has been expanded to include four new classes — *LongAccumulator*, *LongAdder*, *DoubleAccumulator* and *DoubleAdder*

#3: Writing functional programs with Java 8 with simple examples.

The `Operate` interface with the annotation **@FunctionalInterface**. This annotation ensures that you can only have a single abstract method (aka **SAM**). You can have additional default and static method implementations. The **default methods** provide default behaviors, and the closures are used to have special behaviors. The **static methods** are used as helper methods.

So, no longer you need to have the pre Java 8 patterns like

Java 8 (24)

- 01: ♦ 19 Java 8 I
- 02: ♦ Java 8 Stre
- 03: ♦ Functional
- 04: ♥♦ Top 6 tips
- 04: Convert Lists
- 04: Understandir
- 05: ♥ 7 Java FP
- 05: ♦ Finding the
- 06: ♥ Java 8 way
- 07: ♦ Java 8 API
- 08: ♦ Write code
- 10: ♦ ExecutorSe
- Fibonacci numb
- Java 8 String str
- Java 8 using the
- Java 8: 7 useful
- Java 8: Different
- Java 8: Does “O
- Java 8: What is c
- Learning to write
- Non-trivial Java 8
- Top 6 Java 8 fea
- Top 8 Java 8 fea
- Understanding J

JVM (6)

Reactive Programn

Swing & AWT (2)

JEE Interview Q&A (3)

Pressed for time? Jav

SQL, XML, UML, JSC

Hadoop & BigData Int

Java Architecture Inte

Scala Interview Q&As

Spring, Hibernate, & I

Testing & Profiling/Sa

Other Interview Q&A 1

Free Java Interview

- ***java.util.Collection*** is an interface and ***java.util.Collections*** is utility class with helper methods
- ***java.nio.file.Path*** is an interface and ***java.nio.file.Paths*** is a helper class.

Thanks to Java 8 as you can have static helper methods.

#4: 7 useful Java 8 miscellaneous additions worth knowing and talking about in the job interviews if asked.

1. ***String.join()*** method, which is an opposite of ***String.split(...)***.
2. ***Comparator*** interfaces have a number of useful methods to sort objects with options to nest multiple fields with ***thenComparing(...)***, and other handy methods like ***nullsFirst()***, ***nullsLast()***, ***naturalOrder()***, ***reverseOrder()***, etc.
3. In Java 8, ***java.util.Optional*** class has been added to deal with optional object references.
4. In cases where you want to detect result overflow errors in int and long, the methods ***addExact***, ***subtractExact***, ***multiplyExact***, and ***toIntExact*** in Java 8, the ***java.lang.Math*** class throws an ***ArithmeticException*** when the results overflow.
5. The ability to lazily read lines from a file with ***Files.lines(...)***.
6. In Java 8, you can repeat the same annotation with the ***@Repeatable*** annotation.
7. Java 8 introduces the “**Type Annotations**“, which are annotations that can be placed anywhere you use a type.

#5: Different ways to sort a collection of objects in pre and post Java 8.

1. Java library ***Comparator*** pre Java 8
2. Apache commons library ***BeanComparator*** pre Java 8
3. Google ***Gauva library*** functional programming style Pre Java 8
4. Java 8 functional programming approach. Concise and powerful.

16 Technical Key Areas

[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 \(1\)](#)

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 \(1\)](#)
- [Spring & Hibernate Tutorials \(1\)](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

100+ Java pre-interview

#6: Java 8 functional programming is not a silver bullet.

1. Where the functional code has increased readability and maintainability
2. Where performance improvement is possible with Fork/Join parallel processing.
3. Where code becomes easier to refactor in the future
4. Where the code is easier to test and debug.

So, OO programming and functional programming can co-exist.

#7: Understanding Java 8 Streams. A stream is an infinite sequence of consumable elements (i.e a data structure) for the consumption of an operation or iteration. Any Collection can be exposed as a stream. The operations you perform on a stream can either be intermediate (map, filter, sorted, limit, skip, concat, substream, distinct, etc) producing another stream or terminal (forEach, reduce, collect, sum, max, count, matchAny, findFirst, findAny, etc) producing an object that is not a stream. Basically, you are building a pipeline as in Unix.

```
ls -l | grep "Dec" | Sort +4n | more
```

#8: Using the Predicate functional interface. The java.util.function package in Java 8 has a number of functional interfaces, and **Predicate** interface is used for filtering objects from a collection.

Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

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

766 views

18 Java scenarios based interview Questions and Answers

400 views

coding tests

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

- ✚ [Can you write code? \(](#)
- ✚ ♦ [Complete the given](#)
- ✚ [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?

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

- ✚ [Career Making Know-](#)
- ✚ [Job Hunting & Resum](#)

001A: ♦ 7+ Java integration styles & patterns

interview questions & answers

388 views

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

295 views

♦ 7 Java debugging interview questions & answers

293 views

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

285 views

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

279 views

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

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

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. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



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](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ Java 8 using the Predicate functional interface

♥ 9 Tips to earn more as a Java (or any) developer? ▶

Posted in Java 8, member-paid

Tags: TopX

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.