

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](#) › [Ice Breaker Interview Q&A](#) › 03: ♦10+ Know your industry

Q&As for the Java developers

03: ♦10+ Know your industry Q&As for the Java developers

Posted on [September 10, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — 2

[Comments](#) ↓

13

Like

Share

Tweet

2

G+1

Share

Insights into what your prospective employers are looking for. What skills are required for you to go places? Do you know your industry well enough to impress your prospective employers?

Prospective employers will be judging you on 4 key aspects

1) Do you have the right technical “know how”?

Can you not only develop solutions to the functional specifications, but also can you build robust solutions by

[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](#)

└─ 01: ♦ 15 Ice breake

└─ 02: ♥♦ 8 real life ex

└─ 03: ♦10+ Know you

└─ 04: Can you think c

└─ 05: ♥ What job inte

└─ 06: ► Tell me abou

└─ 07: ♥ 20+ Pre inter

☐ [Core Java Interview C](#)

☐ [Java Overview \(4\)](#)

└─ 01: ♦ ♥ 17 Java c

asking the right questions relating to non-functional requirements. For example,

Is this information sensitive enough to require encryption? Will my stake holder be happy with the response time of 2 seconds? Is there a potential race condition here? Should these transactions execute atomically? Will my solution scale if the number of concurrent users are doubled?

2) Will you be interested in the company, project, role and the technologies used? What are your short and long term objectives? What are you expecting in your next role? What motivates you?

3) Will you fit in with the team? Do you get easily annoyed? Do you handle criticisms well? Will you be interested in learning a new language, framework or tool? What is your view on TDD, BDD, continuous integration, etc? What is your take on agile development practices?

4) Do you have the right skills and the attitude to get the job done? Do you get frustrated when the requirements are not clear or take the initiative to get them clarified? Do you prefer being glued to the keyboard and churn out code or taking part in the team sharing sessions to understand the real business driver and propose more effective solutions? Will you be able to communicate your thoughts effectively?

Q1. Why are you leaving your current job? What do you like and dislike about your current job?

A1. Firstly, never be negative about your current or previous jobs. Also, don't bring salary into the discussion.

Likes:

- The team of 6 with varying strengths complimented each other to get the job done, and learned a lot from each other.

02: ♥♦ Java Con

03: ♦ 9 Core Jav

04: ♦ Top 10 mos

☐ Data types (6)

01: Java data ty

02: ♥♦ 10 Java S

03: ♦ ♥ Java aut

04: Understandir

05: Java primitiv

Working with Da

☐ constructors-methc

Java initializers,

☐ Reserved Key Wor

♥♦ 6 Java Modifi

Java identifiers

☐ Classes (3)

♦ Java abstract c

♦ Java class loa

♦ Java classes a

☐ Objects (8)

► Beginner Jav

♥♦ HashMap & H

♦ 5 Java Object i

♦ Java enum inte

♦ Java immutabl

♦♥ Object equals

Java serialization

Mocks, stubs, dc

☐ OOP (10)

♥ Design princip

♦ 30+ FAQ Java

♦ Why favor cor

08: ♦ Write code

Explain abstracti

How to create a

Top 5 OOPs tips

Top 6 tips to go a

Understanding C

What are good r

☐ GC (2)

♦ Java Garbage

- Enjoyed solving business problems and learning new languages (e.g. Groovy & Scala), paradigm (e.g. Functional Programming), technologies (e.g. Apache Spark & Hadoop), and tools (e.g. Cloudera Manager Server and AgileCraft to manage the Scrum process)
- Took pride in identifying gaps and getting them rectified in a collaborative manner.
- Pro-actively identified and fixed performance issues using JMeter, VisualVM, thread dumps, and GC logs.
- Increasing the unit tests coverage to 85% with Mockito for mocking, SonarQube for coverage, JBehave for BDD, JMeter for load testing, and “Metrics Core” for capturing through-put and latencies.

Dislikes:

Reasons for leaving:

- Not challenged enough at my current job. The company offered a great opportunity as someone with 2-3 year experience, and appreciate the skills I acquired there, but now with 5 year hands-on experience, and having completed 6 projects through the full SDLC, I am ready to take on more challenges.
- Looking forward to expanding my domain knowledge in finance, especially in “high frequency trading”.
- Looking for an opportunity to get more involvements in mentoring & applying my leadership skills.

Q2. What was the biggest accomplishment and failure in your current or past jobs?

A2. Reflect back on your career achievements and provide quantified answers.

Achievements:

- Successfully completed a full stack online trading system written in Java 8, Spring, and AngularJS, which handles 400+ concurrent users and integrates with 4 other back-end systems .

03: Java GC tun

Generics (5)

♥ Java Generics

♥ Overloaded m

♦ 12 Java Gener

♦ 7 rules to reme

3 scenarios to ge

FP (8)

01: ♦ 19 Java 8 I

02: ♦ Java 8 Stre

03: ♦ Functional

04: ♥♦ Top 6 tips

05: ♥ 7 Java FP

Fibonacci numb

Java 8 String str

Java 8: What is c

IO (7)

♥ Reading a text

♦ 15 Java old I/C

06: ♥ Java 8 way

Processing large

Processing large

Read a text file f

Reloading config

Multithreading (12)

01: ♥♦ 15 Beginn

02: ♥♦ 10+ Java

03: ♦ More Java

04: ♦ 6 popular J

05: ♦ How a thre

06: ♦ 10+ Atomic

07: 5 Basic multi

08: ♦ ThreadLoc

09: Java FutureT

10: ♦ ExecutorSe

Java ExecutorSe

Producer and Co

Algorithms (5)

♦ Splitting input t

♦ Tree traversal :

♥ ♦ Java coding

- Designed and developed a Java based non-blocking server that communicates with 4500+ EFTPOS outlets with a latency of 20ms
- Spearheaded a “Quick-Win” program that improved the site ranking from 24th to 9th within 5 months in terms of user experience, responsiveness, and reliability.

Failures:

We all learn more from our mistakes or failures than from our successes or achievements. When you mention your mistakes or failures, make it a point to mention what you had learned from those mistakes.

- The team was using a cut down database in the development environment and the technical solution I provided worked well for the low volume of data, but when was moved to production like data, it caused some performance issues. Learned a valuable lesson of validating the solution with more production like data set early on in the SDLC. **I quickly provided the revised & tuned solution within a week and got the the big thumbs up from the testers.**
- Wrote some JUnit based integration tests that were bit fragile due to data fluctuations. The failing tests were causing the builds to fail. Took the initiative to fix this problem by performing the integration tests via more stable data sets that were populated via DBUnit during the set up phase and the data were removed during the tear down phase. Also, introduced other strategies for integration tests by using in memory databases like HSQL DB as opposed to the actual database.

Q3. How do you keep your knowledge current?

A3. Via online articles, blogs, and books.

- <http://www.infoq.com>
- <http://java.dzone.com>
- <http://www.theserverside.com>

Searching algori

Swapping, partiti

Annotations (2)

8 Java Annotatio

More Java annot

Collection and Data

Find the first ne

Java Collection

Java Iterable \

HashMap & H

Sorting objects

02: Java 8 Stre

04: Understandi

4 Java Collection

If Java did not ha

Java 8: Different

Part-3: Java Tre

Sorting a Map by

When to use whi

Differences Between

Java Iterable \

Multithreading

Why do Proxy,

Core Java Modif

Differences betw

Java Collection i

Event Driven Progr

Event Driven Pr

Event Driven Pr

Exceptions (2)

Java exception

Top 5 Core Java

Java 7 (2)

Java 7 fork and j

Java 7: Top 8 ne

Java 8 (24)

01: 19 Java 8 I

02: Java 8 Stre

03: Functional

04: Top 6 tips

04: Convert Lists

- YouTube videos by Venkat Subramaniam, Martin Fowler, etc and following the channels like “Java Brains”, “Cave Of Programming”, etc.
- Following Twitter handles like “@BrianGoetz”, “@joshbloch”, “@AdamBien”, etc.

Q4. How do you know what technologies, frameworks or tools are in demand?

A4. Some technologies/frameworks don't even make it to the mainstream. So, do your research before learning.

1. Through online job portals like “Indeed.com”, “IT Jobs watch”, “Glassdoor.com”, <http://www.google.com/trends>, GIT commits, Stack overflow tags, etc to understand what the prospective employers are looking for? what problems are they trying to solve? what are the modern trends? E.g. ETL vs ELT, MVC vs MVW, BigData vs Data Warehouse, Actor based concurrency models? and so on.

2. Through personal interests, experience and networking with the fellow professionals via LinkedIn. For example, fascination for writing concurrent programs due to the challenges the concurrency presents and tuning the design, code & the JVM to achieve better throughput & lower latency.

Q5. How would you go about learning a new piece of technology, framework, or a tool?

A5.

STEP 1: Get started with good online tutorials. Google for relevant tutorials, but there are a number of good tutorial sites like

- <http://www.vogella.de/java.html>
- <http://www.mkyong.com>

STEP 2: Read online articles, blogs, and books about a particular framework, technology, or tool to understand its high level architecture, core concepts, best practices, and potential pitfalls.

[04: Understanding](#)

[05: ♥ 7 Java FP](#)

[05: ♦ Finding the](#)

[06: ♥ Java 8 way](#)

[07: ♦ Java 8 API](#)

[08: ♦ Write code](#)

[10: ♦ ExecutorSe](#)

[Fibonacci numbe](#)

[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 \(6\)](#)

[♦ Java Garbage](#)

[01: jvisualvm to](#)

[02: jvisualvm to](#)

[05: Java primitiv](#)

[06: ♦ 10+ Atomic](#)

[5 JMX and MBes](#)

[Reactive Program](#)

[07: Reactive Pro](#)

[10: ♦ ExecutorSe](#)

[3. Multi-Threadir](#)

[Swing & AWT \(2\)](#)

[5 Swing & AWT i](#)

[Q6 – Q11 Swing](#)

[JEE Interview Q&A \(3](#)

[JEE Overview \(2\)](#)

[♦ 8 Java EE \(aka](#)

[Java EE interview](#)

[Web basics \(8\)](#)

[01: ♦ 12 Web ba](#)

[02: HTTP basics](#)

[03: Servlet inter](#)

- <http://www.java-tips.org>
- <http://www.javapractices.com/home/HomeAction.do>

STEP 3: It is a best practice to always keep the Java Docs handy.

STEP 4: Watch a few YouTube videos on the topic to further improve the understanding.

STEP 5: If you still have any doubts to be clarified, check with your mentors and networked fellow professionals via LinkedIn.com and forums like JavaRanch.com, StackOverflow, etc.

STEP 6: Find a way to apply or use a particular framework, technology, or tool in a commercial, self-taught, or open-source project. Nothing beats getting **hands-on**. All your previous efforts will go in vain if you don't apply.

STEP 7: Add this particular framework, technology, or tool to your resume's or CV's skills and experience section. Also, blog about your experience as you not only learn more by blogging as you have to research more, but also it will be very useful as your future reference material and for increasing your employ-ability by showing off your **passion** and the technical **know how**, only if done uniquely & properly.

You can explore one of the following blogging tools.

- <http://www.wordpress.org>
- <http://drupal.org>
- <http://www.joomla.org>
- <http://www.blogger.com>

Q6. Why do you think good programmers are often lazy?

A6. Good programmers

- hate repetitive and monotonous tasks. They find the right tools and ways to automate these monotonous and repetitive tasks.

04: JSP overview

05: Web patterns

06: ♦ MVC0, MV

07: When to use

08: Web.xml inte

WebService (11)

01: ♥♦ 40+ Java

02: ♦ 6 Java RE

03: ♥ JAX-RS hc

04: 5 JAXB inter

05: RESTful We

06: RESTful Wel

07: HATEOAS R

08: REST constr

09: 11 SOAP We

10: SOAP Web

11: ♥ JAX-WS hc

JPA (2)

10: Spring, Java

8 JPA interview

JTA (1)

JTA interview Q&

JDBC (4)

♦ 12 FAQ JDBC

JDBC Overview

NamedParamete

Spring, JavaCon

JMS (5)

♦ 16 FAQ JMS ir

Configuring JMS

JMS versus AM

Spring JMS with

Spring JMS with

JMX (3)

5 JMX and MBe

Event Driven Pr

Yammer metrics

JNDI and LDAP (1)

JNDI and LDAP

Pressed for time? Jav

Job Interview Ice B

- don't reinvent the wheel. They will first look for the right tool, API, library, or framework to get the job done with minimal effort.
- constantly learn and refactor the code with good code coverage. They start to dislike the code that they wrote an year ago.
- ask the right questions. Is this code thread-safe? What if an exception is thrown here? Should this class be immutable? Is this class open for extension, but closed for modification? Would this logic require automated unit tests only, integration tests only, or both? Should this config be externalized? Would this be a good candidate to be written as a functional program?

Q7. What qualities do you look for in an effective programmer? How do you recognize a good programmer?

A7.

General Traits

Passion, continuous learning, taking pride in their achievements, ability to look at the big picture and pay attention to details, right attitude, and ability to communicate their thoughts clearly. For example, talk to the business and the stake holders without any technical jargon. Most importantly, getting things done.

Specific Traits

1) Ability to write quality code with best practices & good test coverage.

2) Ability to break a complex problem into manageable chunks with relevant conceptual design diagrams, UML diagrams, ER diagrams, algorithms, and pseudo-code

3) Ability to think and ask the right questions in terms of the 16 key areas?

01: ♦ 15 Ice bre
02: ♥♦ 8 real life
03: ♦10+ Know y
FAQ Core Java J
♥♦ Q1-Q10: Top
♦ Q11-Q23: Top
♦ Q24-Q36: Top
♦ Q37-Q42: Top
♦ Q43-Q54: Top
01: ♥♦ 15 Beginr
02: ♥♦ 10+ Java
FAQ JEE Job Inter
♦ 12 FAQ JDBC
♦ 16 FAQ JMS ir
♦ 8 Java EE (aka
♦ Q01-Q28: Top
♦ Q29-Q53: Top
01: ♦ 12 Web ba
06: ♦ MVC0, MV
JavaScript mista
JavaScript Vs Ja
JNDI and LDAP
JSF interview Q
JSON interview
FAQ Java Web Ser
01: ♥♦ 40+ Java
02: ♦ 6 Java RE
05: RESTFul We
06: RESTful Wel
09: 11 SOAP We
Java Application Ar
001A: ♦ 7+ Java
001B: ♦ Java arc
04: ♦ How to go
Hibernate Job Inter
01: ♥♦ 15+ Hiber
01b: ♦ 15+ Hiber
06: Hibernate Fil
8 JPA interview c
Spring Job Intervie
♦ 11 Spring boot

Q8. What do you understand by the term “hidden experience”?

A8. Good programmers are mainly self-taught and they acquire lots of so called “hidden experience” through proactive and continuous learning and helping others solve their problems. If you just rely only on your experience alone, it can take a lot longer to learn the “**key areas**” and the “**basic concepts**” of programming. It is imperative that you bring out your “**hidden experience**” in your CV and at job interviews.

Q9. When you are coding, what documentation do you have handy?

A9.

1. The APIs for the relevant technologies used. For example Java API, Enterprise Java API, Spring API, jQuery API, JavaScript API, etc. These APIs can also be googled at will with the right search keys like “jQuery API”, “Java 6 API”, etc.

2. The relevant reference manual and home web sites bookmarked for the relevant technologies. The home sites can be Googled with right keywords like “Spring framework home”, “jQuery home”, etc and the reference manual can be Googled with keywords like “Spring framework 2.0 reference”, etc.

3. Often not knowing the key terms for a particular piece of technology is the major challenge for the beginners of a particular technology. This is where the “**cheat sheets**” come in very handy. You can find a number of handy cheat sheets from [cheat sheets](#)

4. Sites like [Java Practices](#) provide good working examples.

Q10. Do you periodically review your strengths and weaknesses? if yes, what are your current weaknesses?

A10. When you mention a weaknesses, make it a point to clarify as to how you are going about fixing improving your weaknesses. You can also flaunt your industry knowledge.

01: ♥♦ 13 Spring

01b: ♦ 13 Spring

04 ♦ 17 Spring b

05: ♦ 9 Spring B

Java Key Area Ess

♦ Design pattern

♥ Top 10 causes

♥♦ 01: 30+ Writir

♦ 12 Java desigr

♦ 18 Agile Develo

♦ 5 Ways to debi

♦ 9 Java Transac

♦ Monitoring/Pro

02: ♥♦ 13 Tips to

15 Security key :

4 FAQ Performa

4 JEE Design Pa

5 Java Concurr

6 Scaling your J

8 Java memory i

OOP & FP Essenti

♦ 30+ FAQ Java

01: ♦ 19 Java 8 I

04: ♥♦ Top 6 tips

Code Quality Job I

♦ Ensuring code

♦ 5 Java unit tes

SQL, XML, UML, JSC

ERD (1)

♦ 10 ERD (Entity

NoSQL (2)

♦ 9 Java Transac

3. Understanding

Regex (2)

♥♦ Regular Expr

Regular Express

SQL (7)

♦ 15 Database d

♦ 14+ SQL interv

♦ 9 SQL scenari

Auditing databas

Example:

- Recently identified that I don't have much knowledge or experience on scripting languages like Perl or Python. Hence, I have recently started working on some Python tutorials from site XYZ.
- JavaScript based frameworks like Angular JS is gaining lots of adoption, so I have started learning JavaScript and Angular JS in my spare time.
- Heard everyone talking about BigData, so downloaded the Cloudera express server on VMware, and wrote some basic Spark jobs to gain more insights into Hadoop.

In some cases a weakness can be construed as a strength

Example:

- When I am faced with a complex problem, I tend to keep at it for a long time until I reach a resolution. This gets me into looking more within the square. Even though I know that taking a break from the problem or chatting to a colleague is more conducive to looking outside the square.

So, I am currently working on taking a break from a complex problem after trying for say 45 minutes or so. Often I get a solution whilst taking a brief walk, commuting to work or explaining the problem to a colleague.

- At times I tend to be less of a listener and more of a talker to add value or contribute. I strongly understand that a good communicator must be a great listener, hence I have been consciously working on this, and recently noticed a significant improvement.

Popular Posts

♦ 11 Spring boot interview questions & answers

Deleting records

SQL Subquery in

Transaction man

UML (1)

♦ 12 UML intervi

JSON (2)

JSON interview (

JSON, Jackson,

XML (2)

XML basics inter

XML Processing

XSD (2)

11 FAQ XSD inte

XSD reuse inter

YAML (2)

YAML with Java

YAML with Sprin

Hadoop & BigData Int

♥ 01: Q1 – Q6 Had

02: Q7 – Q15 Hado

03: Q16 – Q25 Hac

04: Q27 – Q36 Apa

05: Q37 – Q50 Apa

05: Q37-Q41 – Dat

06: Q51 – Q61 HB

07: Q62 – Q70 HDI

Java Architecture Inte

♥♦ 01: 30+ Writing

001A: ♦ 7+ Java int

001B: ♦ Java archi

01: ♥♦ 40+ Java W

02: ♥♦ 13 Tips to w

03: ♦ What should I

04: ♦ How to go ab

05: ETL architectur

1. Asynchronous pi

2. Asynchronous pi

Scala Interview Q&As

01: ♥ Q1 – Q6 Scal

02: Q6 – Q12 Scal

03: Q13 – Q18 Sca

852 views

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

823 views

18 Java scenarios based interview Questions and Answers

447 views

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

400 views

♦ 7 Java debugging interview questions & answers

311 views

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

301 views

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

291 views

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

285 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

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.



04: Q19 – Q26 Sca

05: Q27 – Q32 Sca

06: Q33 – Q40 Sca

07: Q41 – Q48 Sca

08: Q49 – Q58 Sca

09: Q59 – Q65 Hig

10: Q66 – Q70 Pat

11: Q71 – Q77 – S

12: Q78 – Q80 Rec

Spring, Hibernate, & I

Spring (18)

Spring boot (4)

♦ 11 Spring bc

01: Simple Sp

02: Simple Sp

03: Spring bo

Spring IO (1)

Spring IO tuto

Spring JavaConl

10: Spring, Ja

Spring, JavaC

Spring, JavaC

Spring, JavaC

01: ♥♦ 13 Spring

01b: ♦ 13 Spring

02: ► Spring DI

03: ♥♦ Spring DI

04 ♦ 17 Spring b

05: ♦ 9 Spring B

06: ♥ Debugging

07: Debugging S

Spring loading p

Hibernate (13)

01: ♥♦ 15+ Hiber

01b: ♦ 15+ Hiber

02: Understandir

03: Identifying ar

04: Identifying ar

05: Debugging H

06: Hibernate Fil

07: Hibernate mi



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.

< EIP

4 JEE Design Patterns Interview Q&As >

Posted in Ice Breaker Interview Q&A, Job Interview Ice Breaker Q&A Essentials, Judging Experience Interview Q&A

2 comments on “03: ♦10+ Know your industry Q&As for the Java developers”



thayap says:

March 31, 2016 at 4:51 am

Priceless and absolutely fabulous article for developers

[Reply](#)



Laxmanrao Potadar says:

March 24, 2015 at 8:27 pm

Its very good article with lot of information

[Reply](#)

Leave a Reply

08: Hibernate au

09: Hibernate en

10: Spring, Java

11: Hibernate de

12: Hibernate cu

AngularJS (2)

♥ 8 AngularJS in

More Angular JS

Git & SVN (6)

♥ Git & Maven fc

♥ Merging Vs rel

♥ Understanding

6 more Git interv

8 Git Source cor

Setting up Cygw

JMeter (2)

♥ JMeter for test

♦ JMeter perform

JSF (2)

JSF interview Q&

More JSF intervi

Maven (3)

♥ Git & Maven fc

12 Maven intervi

7 More Maven ir

Testing & Profiling/Sa

Automation Testing

♥ Selenium and

Code Coverage (2)

Jacoco for unit te

Maven and Cobr

Code Quality (2)

♥ 30+ Java Code

♦ Ensuring code

jvisualvm profiling (

01: jvisualvm to :

02: jvisualvm to :

03: jvisualvm to :

Performance Testir

♥ JMeter for test

♦ JMeter perform

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

Comment

Post Comment

[Unit Testing Q&A \(2\)](#)

[BDD Testing \(4\)](#)

[Java BDD \(Be](#)

[jBehave and E](#)

[jBehave and j](#)

[jBehave with t](#)

[Data Access Uni](#)

[♥ Unit Testing](#)

[Part #3: JPA H](#)

[Unit Test Hibe](#)

[Unit Test Hibe](#)

[JUnit Mockito Sp](#)

[JUnit Mockito](#)

[Spring Con](#)

[Unit Testing](#)

[Part 1: Unit te](#)

[Part 2: Mockit](#)

[Part 3: Mockit](#)

[Part 4: Mockit](#)

[Part 5: Mockit](#)

[Testing Spring T](#)

[Integration Un](#)

[Unit testing Sp](#)

[♦ 5 Java unit tes](#)

[JUnit with Hamc](#)

[Spring Boot in u](#)

[Other Interview Q&A 1](#)

[Finance Domain In](#)

[12+ FX or Forex](#)

[15 Banking & fin](#)

[FIX Interview Q&A](#)

[20+ FIX basics in](#)

[Finding your way](#)

[Groovy Interview Q](#)

[Groovy Coding C](#)

[Cash balance](#)

[Sum grades C](#)

[♥ Q1 – Q5 Groov](#)

[♦ 20 Groovy clos](#)

[♦ 9 Groovy meta](#)

[Groovy method c](#)

- Q6 – Q10 Groov
- JavaScript Interview
- JavaScript Top I
- ♥ Q1 – Q10 J
- ♦ Q11 – Q20
- ♦ Q21 – Q30
- ♦ Q31 – Q37
- JavaScript mis
- JavaScript Vs Ja
- JavaScript Vs
- Unix Interview Q&A
- ♥ 14 Unix intervi
- ♥ Hidden Unix, C
- sed and awk to v
- Shell script inter
- Unix history com
- Unix remoting in
- Unix Sed comm
- Free Java Interview
- ▶ Java Integration
- ▶ Java Beginner I
- 02: ▶ Spring DIP, I
- 06: ▶ Tell me abou

As a Java Architect

[Java architecture & design concepts](#)
[interview Q&As with diagrams](#) | [What should be a typical Java EE architecture?](#)

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 \(1\)](#)

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 \(45\)](#)

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

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.