

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](#) › 01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

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

Posted on [August 29, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — [No Comments](#) ↓



Tweet



2

Share

Q1. Tell us about yourself?

This question is asked 95% of the time.

[Hint: Focus on your strengths and marry up your strengths with the job specification. For example,

[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

- 5 year hands on experience in building web based applications to integrating disparate systems using web services, messaging, and XML/JSON based data formats.
- Completed 3 projects through full SDLC process using sought-after technologies X, Y, and Z, and frameworks P and Q in 2 of those projects. One of the projects was a mission critical transactional system that handles 300 concurrent users;
- Enhanced my skills in SOA based architectures and standalone scheduled batch jobs using bulk loads and ETL operations with IBM Datastage at XYZ Bank.]
- Initiation of any “**QuickWins**” projects. The focus is to improve the overall effectiveness and usefulness of a system through small changes in a collaborative effort with the business. For example, automating tedious manual tasks, improving website user experience/rankings, stabilizing the environment, etc.
- **Fixed security holes.** Security is of paramount importance to any application or website. Applications with security vulnerabilities can not only tarnish the reputation of a company, but also can adversely impact the bottom-line of that organization.
- **Isolated and reproduced intermittent issues.** Intermittent issues are hard to reproduce and debug. A novice developer will promptly mark those defects in the bug tracking system as “cannot be reproduced” without having the skill to isolate, reproduce and then fix the issue. An experienced developer with good grasp on the key areas will not only have will not only have the competency to isolate, reproduce, and fix these issues, but also will have the skill to identify these potential issues during code review.
- Acted as a **change agent** to enforce standards and improve processes. A change agent not only changes a system’s or application’s behavior but also people’s behavior, attitude, and culture towards writing quality software. People’s

- 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

behavior can be changed by enforcing good coding standards. Substandard software development processes can also demotivate developers and adversely impact their behavior and attitude. The development processes need to be continuously improved by adopting good tools and practices.

— **Identified performance issues and bench marked** them with the view to improve. I have never been in a project or organization that is yet to have any performance or scalability issues. It is a safe bet to talk up your achievements in fixing performance or resource leak issues in job interviews. Unlike web security testing, performance testing is very prevalent in many applications. Premature optimization of your code is bad as it can compromise on good design or writing maintainable and testable code. But one needs to be aware of potential causes for performance issues that can occur due to major bottlenecks in a handful of places or minor inefficiencies in thousands of places (i.e. death by thousand cuts).

— **Championed** continuous improvement programs. Organizations must develop a culture of continuous improvement in order to flourish. I am yet to work for an organization that did not have any environmental or process related issues and challenges. Many of these tasks can be simplified, automated, and integrated to avoid monotonous and error prone human intervention. So, experience with the agile practices, continuous integration and build, environmental improvements, etc will be of great asset to any organization. A software must be architected for **deployability**. Deployability is a non-functional requirement that addresses how reliably and easily a software can be deployed from development into the production environment.

— Assessed **non-functional requirements**. Like deployability, many non-functional requirements like security, performance, transaction management, thread-safety, quality of service, etc are often neglected or overlooked during application development. Inexperienced or rushed developers can create systems that work great in development, but have

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

many problems in the test or production environments due to sheer load, concurrent access, locale or timezone settings, portability issues due to operating system specific file separator or new line character, hard coded values requiring externalization, etc. If you identify any gaps in these non-functional requirements, take the initiative to close the gap as you would do with any functional requirements.

— A **pragmatic person** with a wide range of experience and skills to know what works for a given environment without all the hype. When designing or developing systems, be pragmatic without all the buzz words and hype. Pick the proven technology and framework stacks that will get the job done. Ensure that you have the right people with the relevant skills in those technology stacks. Evaluate the architecture and choice of technology/framework/tool with a proof of concept. Determine if a particular architecture, process, or methodology will work for your environment and culture. For example, SOA is an enterprise architecture strategy where business logic is organized into services, and business processes are realized through the invocation of these services. SOA has benefits such as bridging the divide between the I.T. and the business, loose coupling, reuse of business logic, etc. But to successfully implement a SOA based architecture, a number of organizational boundaries need to be bridged. This typically requires a change in organizational culture, which is very difficult to achieve in some organizations that are siloed and have different business teams looking after different products. There are also other inherent challenges like changing a service implementation can break one or more of the consumers, overall reliability of the system when one service is down, rolling back other services when a service fails, propagating security and transactional contexts between the services, deployability of those services and performance considerations in devising services as too many remote calls can adversely impact performance. At times you need to come up with a hybrid approach to attack a problem. The software development is full of trade-offs and be pragmatic about getting things done by looking at the big picture and having attention to detail. For example, an organization using

[Searching algorithm](#)
[Swapping, partitioning](#)
[Annotations \(2\)](#)
[8 Java Annotations](#)
[More Java annotations](#)
[Collection and Data Structures](#)
[♦ Find the first non-repeating character in a string](#)
[♦ Java Collections](#)
[♥ Java Iterable & Iterator](#)
[♥♦ HashMap & HashSet](#)
[♦ Sorting objects](#)
[02: ♦ Java 8 Streams](#)
[04: Understanding Java 8 Streams](#)
[4 Java Collections](#)
[If Java did not have Collections](#)
[Java 8: Differences between Java 7 and Java 8](#)
[Part-3: Java Tree Traversal](#)
[Sorting a Map by Value](#)
[When to use which Collection](#)
[Differences Between Java 7 and Java 8](#)
[♥ Java Iterable & Iterator](#)
[♦ Multithreading](#)
[♦ Why do Proxy, Stub, and Decorator patterns exist?](#)
[Core Java Modifications in Java 8](#)
[Differences between Java 7 and Java 8](#)
[Java Collections](#)
[Event Driven Programming](#)
[Event Driven Programming](#)
[Event Driven Programming](#)
[Exceptions \(2\)](#)
[♦ Java exceptions](#)
[Top 5 Core Java Collections](#)
[Java 7 \(2\)](#)
[Java 7 fork and join](#)
[Java 7: Top 8 new features](#)
[Java 8 \(24\)](#)
[01: ♦ 19 Java 8 Interview Questions](#)
[02: ♦ Java 8 Streams](#)
[03: ♦ Functional Programming](#)
[04: ♥♦ Top 6 tips for Java 8](#)
[04: Convert Lists to Arrays](#)

a waterfall approach to software development could adopt some of the agile practices to suit their environment.

Q2. Tell us about the most satisfying project you worked on in the last year? What were your contributions?

[Hint: Rewriting the whole batch process framework that not only used to take 17 hours to run but also was very hard to maintain as changing one part of the code can break another part of the application. The re-written batch framework with modular design, multi-threading, distributed caching, tuned SQL statements, and revised database tables completes in under 3 hours. My contributions involved designing the solution and leading the development.]

Q3. What are the challenges you faced both technically and non-technically in your past assignments?

[Hint: Fixing performance, thread safety, transactional issues and security holes, spearheading process improvements, taking the initiatives to close the gaps in business requirements, convincing the management with the open-source adoption, streamlining the release and deployment processes, and getting the business users to agree on alternative solutions.]

Q4. Why do you like software development?

[Hint: Passion for building things, and satisfaction you get in seeing all the functional requirements and non-functional technical key areas like design, design patterns, thread safety, performance, security, code quality, etc fit together like a puzzle. Solving problems relating to above mentioned key areas gives me a sense of accomplishment. Like the fact that one needs to look at the big picture as well as pay attention to details. It is also quite interesting to look at things from both

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 s

02: jvisualvm to c

05: Java primitiv

06: ♦ 10+ Atomic

5 JMX and MBe

☐ 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

business/end user perspective (e.g. user interface design, ease of use, how the business operates) and technical perspective. Derive pleasure in working with people to get things done in a team environment by collaborating, mentoring, and facilitating to get things over the line.]

Q5. What do you do when the requirements are not clear?

[**Hint:** take initiatives to get it clarified and properly documented. Conduct workshops, meetings, and brainstorming sessions to gather requirements. Clearly document your assumptions, potential risks, impacts, and mitigation strategies.]

Q6. How would you proceed if you had been assigned a task that involves an unknown technology or business domain?

[**Hint:** Learn by doing a quick tutorial and grasping its fundamentals. Seek others' help. Take it as a challenge. For example, had to collaborate with the access control team that looks after Siteminder to get SSO (Single-Sign-On) working for our application.]

Q7. In your experience, what are some of the harder issues to resolve, and how would you go about resolving them?

[**Hint:** Thread-safety and transactional issues. A better candidate will also touch on non-technical challenges like effectively managing changing requirements, proactively gathering requirements, and taking the initiatives to closing the gaps in requirements and technical design. They may also touch on implementing agile practices, recognizing the importance of collaborative effort, following up with the business for clarification or convincing the business with simpler and more effective solutions to meet their requirements. For example, identified an intermittent

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 c

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 AMC

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

database deadlock issue by emulating user experience under load, and fixed it by adding deadlock retry service interceptors to address intermittent database lockouts and service outages.]

Q8. Describe the types of teams you have worked in, and tell me what worked well and what did not?

[Hint: Team with generalists and specialists who complement each other to achieve a common goal. Team that engages relevant agile practices for better communication and corporation. Well balanced team with right technical skills and good soft skills. Team that hires good caliber people via rigorous screening process.

A team that blame each other, a team that does not communicate well with each other, a team that says “I did it” instead of “we did it”, etc]

Q9. What was the hardest “sell” of an idea or method you have had to make to get it accepted?

[Hint: Adoption of open-source technologies and agile development practices. Quick win strategies to enhance user experience and website rankings. Convinced the business and the management to rewrite an application as opposed to providing a band-aid solution.

Good programmers are open to new ideas and constructive criticisms. Most importantly communicate their ideas effectively in simple terms with examples, mockups, demonstration, diagrams, and analogy]

Q10. Can you draw me a high level architecture of the system you were involved in your last or current position?

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 Interview

♦ 11 Spring boot

[Hint: Good candidates are careful to explain things well in simple terms that even a non-technical person can understand. They will start by drawing a simple box diagram of the architecture and elaborate it further based on their target audience – business goals achieved, design alternatives, design decisions, lessons learned, protocols & data formats used, concepts & patterns applied, development methods used, choice of frameworks, etc.]

Q11. Explain how application you built added value to your business and what was the business problem that it solved?

[Hint: Better customer experience by fully automating a number of manual and partially automated systems. Less error-proneness and much shorter lead times leading to more loyal customers, and much happier support staff.]

Q12. Tell me about an instance where you had to solve a complex design issue and how did you go about solving it?

[My experience: A customer order with a complex life cycle of state changes like pending, authorized, completed, canceled, amended, partially filled, rejected, purged, etc had to be managed properly by capturing the state changes. A state chart diagram was created to capture and clarify things. Broke the problem into more manageable tasks. Did my own research to get a better understanding and others' perspectives on the problem. Used process flow, sequence, and state diagrams to get better clarity and presented it effectively to others to ensure that we are on the same page. Prepared a matrix highlighting all possible "what if" scenarios. Wrote down the possible solutions. Analyzed the pros and cons of each solution. Listed how my solution addressed each scenario.]

Q13. Have you ever been assigned a task of optimizing an application,

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

and how did you go about it?

[My experience: Complex online reports generation clogged up the application with high CPU usage. Converted the report generation process to be asynchronous by submitting the report requests captured via online GUI to a queue, and having a separate process pickup from the queue to generate the report asynchronously and make the report available either via emails or file downloads. The report generation failures are reported via emails.]

[Hint: Understand user behaviors and gather load requirements. Monitor the application and gather metrics. Identify and isolate any similar patterns or behaviors. Simulate the load in a non-production environment using tools like JMeter to gather benchmarks. Profile the application under simulated load. Identify the bottlenecks. Fix the bottlenecks through relevant design changes, configuration changes, and code changes. Run your simulated load tests again, and compare the results against the benchmarks. Document the results for future tuning and reference.]

Q14. What are the critical factors you look for in evaluating the work of others?

[Hint: Ability to look at the bigger picture and drill down to details. Ability to get things done methodically by keeping things simple and asking the right questions – what if an exception is thrown?, What if this method gets accessed by multiple-threads?, etc. Passion to write quality code by constantly improving on the “first-cut” or draft code with proper test cases. Motivation to use code quality and unit test coverage tools like Sonar.]

Note: Remember the fact when you talk about others, the interviewers are thinking about your suitability.

Q15. Describe a situation in which you had to solve a problem without having all the information you

Deleting records
SQL Subquery ir
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 Hadc
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 archil
01: 40+ Java W
02: 13 Tips to w
03: What should l
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

needed – what did you do and what happened?

[**Hint:** In larger organizations, knowledge and expertise are scattered. Collaborate within and outside your team to gather relevant information. Organize meetings and send emails requesting for relevant documentation. Conduct work shops and brainstorming sessions to pick others' brains. Come up with all possible causes. A cause effect diagram can come in handy for more complex problems. Identify and isolate possible causes, and try to reproduce it by emulating the real life scenarios with appropriate load.]

Popular Posts

♦ 11 Spring boot interview questions & answers

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

284 views

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

263 views

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

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.



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 development process interview Q&A

♦ 7 Java debugging interview questions & answers >

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

Tags: Architect FAQs, Core Java FAQs, Java/JEE FAQs, JEE FAQs

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 Groovy
- JavaScript Interview
- JavaScript Top Interview Questions
- ♥ Q1 – Q10 JavaScript Interview Questions
- ♦ Q11 – Q20 JavaScript Interview Questions
- ♦ Q21 – Q30 JavaScript Interview Questions
- ♦ Q31 – Q37 JavaScript Interview Questions
- JavaScript miscellaneous
- JavaScript Vs Java
- JavaScript Vs Python
- Unix Interview Q&A
- ♥ 14 Unix interview questions
- ♥ Hidden Unix, C, sed and awk to v
- Shell script interview
- Unix history com
- Unix remoting in
- Unix Sed comman
- Free Java Interview
- ▶ Java Integration
- ▶ Java Beginner Interview
- 02: ▶ Spring DIP, I
- 06: ▶ Tell me about

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 Tutorial \(1\)](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Tutorial \(1\)](#)
- [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.