

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](#) › 02: ♥♦ 8 real life examples of SAR technique for Java developers

02: ♥♦ 8 real life examples of SAR technique for Java developers

Posted on [September 7, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — 1

[Comment](#) ↓

13

Tweet

2

2

The open-ended questions give you a great opportunity to promote your strengths and accomplishments. SAR (Situation-Action-Result) technique examples.

Example 1: Tuning performance

Situation: Performance problem where the application server had to be restarted every day.

[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](#)

Action:

- Used [JMeter](#) to simulate the load conditions and reproduce the issue.
- Identified the cause of the problem to be leaking database connections due to not properly closing the connections under an exceptional scenario. Used the profiling tool “[VisualVM](#)”. [jvisualvm to detect memory leak](#)
- Fixed the issue by closing the database connections in the finally block.
- Tuned the JVM settings and configured proper service timeouts.
- Load and endurance tested the fixed code with the load testing tool JMeter to confirm that the issue has been fixed.

Result: The application became a true mission critical 24×7 type with a much improved performance.

Example 2: Code quality

Situation: Java code that is hard to maintain and reuse. Changes to one module may break another module.

Action:

- Wrote unit tests with proper mock objects for the existing un-maintainable code.
- Introduced [SonarQube](#) to ascertain code coverage & get code quality metrics and fix the blocker, critical, and major severity items were fixed.
- Re-factored the code with OO concepts and design patterns in a test driven manner to improve maintainability.
- Large procedural style if/else statements were replaced with objects adhering to the Open-Closed design principle.
- Code duplication was eliminated with the help of SonarQube tool.
- Reran the unit tests to ensure that the functionality is not broken due to refactoring.

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 loac

♦ 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

Result: The application became easier to maintain, extend, and reuse. The code coverage was increased from 27% to 76%.

Example 3: Quick wins

Situation: The financial service websites are ranked by an independent body, and this particular company's website that I was the technical lead for was ranked 23rd out of 31 possible companies that took part.

Action: I took the initiative with the collaboration of the business and technical leaders to launch a "QuickWins" project to improve the overall ranking of the website. An independent user experience consultant was hired to analyze and produce a report with 18 most important things that can potentially improve the user friendliness, look and feel, and ease of use of the overall website. Out of those 18 recommendations, 4 of them needed major design and development changes, and did not stack up well in the cost-benefit analysis. The remaining 14 recommendations were implemented within 3 months. The implementation was fast-tracked by adopting some of the agile development practices like iterative development, daily stand-up meetings, and regular catch-ups with the business.

Result: This initiative was a major success and the website ranking was improved from being 23rd to 12th. The management was very impressed, and the contributions were well noticed and rewarded. That was also one of my longest and rewarding contracts.

Example 4: Concurrency Management

Situation: The production ready application consumed very less CPU and response times were very poor due to heavy I/O operations like database read/write operations

Action:

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

- Monitored the CPU usage with Visual VM tool.
- Got a series of thread dumps, say 7 to 10 at a particular interval, say 5 to 8 seconds and analyzed those thread dumps by importing the thread dumps into “Samurai”.
- Paid attention to the blocked threads in red. Alternatively, [VisualVM is handy for debugging deadlocks analyzing thread dumps](#).
- Fixed the concurrency issue by reducing the synchronization granularity in the code.
- The offending SQL statement was identified with a query planner and tuned.

Result: The response times were improved by 200% and CPU usage increased from 45% to 88%. The response times were halved.

Example 5: Design

Situation: The business wants to review the business rules and rapidly change them as the government compliance requirements like tax law changes.

Action:

- Used Drools (open-source rules engine) to externalize the rules to a rules database table.
- A Web based user interface was provided to the business to modify the rules in the database table.
- An approval process was designed to enable a separate user to authorize the change before it takes effect.
- Bulk changes to the rules can be carried out by the operations staff updating the rules table via SQL, but it needs to be tested and signed-off in a staging environment first.

Result: The business was empowered to react rapidly to the regulatory changes from the government. When the regulatory requirements change, the rules within the database tables can be modified in a controlled manner without requiring code changes in most cases.

[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: Different ways to iterate over a collection](#)
[Part-3: Java Tree Traversal](#)
[Sorting a Map by value](#)
[When to use which Collection](#)
[Differences Between Java Collections](#)
[♥ Java Iterable & Iterator](#)
[♦ Multithreading](#)
[♦ Why do Proxy, Decorator, Adapter, Strategy, Composite, Singleton, Factory, Builder, etc. exist?](#)
[Core Java Modifications](#)
[Differences between Java Collections and Java Streams](#)
[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](#)

Drools Tutorials – Non-trivial examples step by step

Example 6: Security

Situation: The SSL handshake issues whilst deploying the application to production environment.

Action:

- Use of keytool to verify certificates in the keystores and truststores
- Exporting the missing certificates from the browser and importing it into the truststore.
- Enabling [SSL debugging](#) and using other handy tools like OpenSSL. The SSLPoke was a very handy Java program to debug SSL issues.

Result: SSL issues were resolved in a timely manner.

Example 7: Technical capabilities & experience

Situation: Develop a stand-alone TCP server that talks to 3000+ petrol pumps as part of the “pay at the pump” solution. The server also needs to integrate with 3+ back-end systems like payment gateway to settle the transactions, loyalty system to accrue & redeem rewards points, data warehouse system to produce 3 dimensional reports, and ESB(i.e. Enterprise Service Bus) to integrate with the ERP system.

Action:

- Developed the stand-alone server using a non blocking I/O framework known as the MINA.
- As a lead developer, had to liaise with internal staff & external vendors from the USA and South Africa. Attended many conferences and one on one sessions to gather functional & non-functional requirements.
- Reduced the complexity of the overall system by revising the architecture & reducing the number of

04: Understanding

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)

♦ Java Garbage

01: jvisualvm to s

02: jvisualvm to c

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

messages flowing among various systems from 89 to just 34.

- Built the system with low latency in mind. [13 Tips to write low latency applications in Java](#)
- Fixed an **intermittent** load balancer issue.
- Managed a team of 7+ developers & liaised with 20+ other stake holders including external vendors, business analysts, systems team & the management.

Result: The high-visibility and low latency application was built on time and within budget with 75% unit test coverage.

Example 8: Facilitating process improvements

Situation: Tasked with transitioning the current project management process from waterfall to **agile**.

Action:

- Took on the role as a facilitator of the **Scrum** agile methodology, and empowered the teams to be more self-organizing
- Promoted Test Driven Development (TDD) and Behavior Driven Development (BDD) practices to increase the automated test coverage from 40% to 70%.
- Mentored the teams in shifting their mindsets from the big bang approach to iterative & incremental approach with the key focus of delivering a business value in each iteration.
- Trained staff on writing epics & stories, sprint planning, breaking down the stories into tasks, sizing the stories & tasks, conducting daily scrum & fortnightly retrospective sessions, velocity points, progress tracking charts like burn-down & burn-up, etc

Result: A customized & practical agile process was embraced by the team that worked well for them without all

[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 S](#)

[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](#)

the rigid rules & rituals, but encompassing the core agile values and principles that added value to the business.

Key Take Away

Sell your strengths and well-rounded abilities to add value to the business. The key traits are: **1)** passion **2)** strong technical skills, **3)** necessary soft skills, and **4) right attitude/**.

So, don't get annoyed when you are technically grilled or challenged. If you don't know or in the wrong accept it. Show your enthusiasm for the job and the chosen career path.

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

285 views

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

263 views

01: ♦ 15 Ice brea

02: ♥♦ 8 real life

03: ♦10+ Know y

FAQ Core Java Jot

♥♦ 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

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.

< 8 Java Annotations interview Questions and Answers

More Java annotations interview questions & answers >

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

One comment on “02: ♥♦ 8 real life examples of SAR technique for Java developers”

Uma says:

September 27, 2016 at 1:25 am

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



This is an awesome article. Very informative and thought provoking. Thank you:)

Reply

Leave a Reply

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

Comment

Post Comment

[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 Hadc](#)
[03: Q16 – Q25 Hac](#)
[04: Q27 – Q36 Apa](#)
[05: Q37 – Q50 Apa](#)
[05: Q37-Q41 – Dat](#)
[06: Q51 – Q61 HBa](#)
[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 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](#)

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 box
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

- 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

☐ 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 ui

☐ 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 In](#)
- [Low Latency \(7\)](#)
- [Memory Management](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

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 Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [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

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.