

Java-Success.com

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

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

[Home](#) › [Interview](#) › [Core Java Interview Q&A](#) › [Reserved Key Words](#) › ♥♦ 6 Java

Modifiers every interviewer seems to like

♥♦ 6 Java Modifiers every interviewer seems to like

Posted on [August 10, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — No

[Comments](#) ↓

7
Like
Share

Tweet

2
G+1

1

Share

Q1. In Java, what purpose does the key words **final**, **finally**, and **finalize** fulfill?

A1. Covered in detail at [Q1-Q10: Top 50+ Core Java Interview Questions & Answers](#)

Q2. What is the difference between '**final**' and '**const**' modifiers on a variable?

A2. This is a bit of a tricky question because the '**const**' is a reserved keyword in Java, but not used. In C++ it means a variable is a constant(i.e. its values cannot be changed).

[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

final

Final modifier on a reference variable just means that the reference cannot be changed to reference a different object once assigned. This does not mean that the variable is a constant because the values of the object it refers to can be modified unless the values themselves are marked as final. For example, an *Employee* object marked as final may have member variables "firstName", and "lastName". The values of "firstName", and "lastName" can be modified if they themselves are not marked as final.

```

1
2  final Employee emp1 = new Employee("John", "Pe
3  emp1 = new Employee(); // compile-time error.
4
5  //if 'firstName' variable in Employee class is
6  emp1.setFirstName("Simon"); //Line A - firstNa
7
8  //if 'firstName' itself marked as final then,
9

```

const

is a reserved keyword in Java to make a variable constant, so that the referenced object values also cannot be modified, but it is currently not used in Java. The compiler will complain if you use it.

Q3. What is a **volatile** key word in Java?

A3. The volatile keyword is used with object and primitive variable references to indicate that a variable's value will be modified by different threads.

Volatile

means

- The value of this variable will never be cached locally within the thread, and all the reads and writes must go to the main memory to be **visible** to the other

02: ♥♦ Java Cor

03: ♦ 9 Core Jav

04: ♦ Top 10 mos

☐ Data types (6)

01: Java data typ

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

threads. In other words the keyword **volatile**

guarantees visibility.

- From JDK 5 onwards, writing to a volatile variable happens before reading from a volatile variable. In other words, the volatile keyword guarantees **ordering**, and prevents compiler or JVM from reordering of the code.

Q4. How does **volatile** keyword differ from the **synchronized** keyword?

A4.

1. The **volatile** keyword is applied to variables of both primitives and objects, whereas the **synchronized** keyword is applied to only objects.
2. The **volatile** keyword only guarantees visibility and ordering, but not atomicity, whereas the **synchronized** keyword can guarantee both visibility and atomicity if done properly. So, the **volatile** variable has a limited use, and cannot be used in compound operations like incrementing a variable.

Wrong use of volatile in a compound operation

```

1  volatile int counter = 0;
2
3  public void increment(){
4      counter++;
5  }
6

```

Right use of volatile. **Example1:**

```

1  volatile boolean status = false;
2
3  //...
4
5  public void process(){
6      while(!status){
7          //....
8      }
9  }
10

```

Or in lazy singleton. **Example2:** Double checked locking

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 Beginr

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

```

1  public final Class MySingleton {
2
3      private static volatile MySingleton instance
4
5      private MySingleton( ){}
6
7      public static MySingleton getInstance() {
8          if(instance == null) {
9              synchronized (MySingleton.class)
10                 if(instance == null) {
11                     instance = new MySingleton();
12                 }
13             }
14         }
15
16         return instance;
17     }
18 }
19
20

```

Important: Synchronized keyword (i.e. locking) can guarantee both visibility and atomicity, whereas volatile variables can only guarantee visibility. A synchronized block can be used in place of volatile but the inverse is not true.

So, if you are not sure where to use, then use the “synchronized” keyword, and stay clear of the volatile modifier. You can learn more in detail at [“10+ Atomicity, Visibility, and Ordering interview Q&As on Java Memory Model \(JMM\) to understand multi-threading”](#)

Q5. What is a “transient” modifier? Can you mark a static variable as transient?

A5. It marks a member variable not to be serialized when it is persisted to streams of bytes. It cannot be used with a static variable as a static variable belongs to a class, not to an object. You can only serialize an object.

Transient

Serialization converts an object state to serial bytes (i.e. flattening an object). Those bytes are sent over the network and the object is recreated from those bytes. Member variables marked by the java transient keyword are not transferred over the wire. A “File” object cannot be serialized.

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 Framework

Java Iterable and Iterator

HashMap & HashSet

Sorting objects

Java 8 Streams

Understanding Java 8 Streams

Java Collections Framework

If Java did not have Collections

Java 8: Different Collections

Part-3: Java Tree Traversal

Sorting a Map by Value

When to use which Collection

Differences Between Java Collections

Java Iterable and Iterator

Multithreading

Why do Proxy, Stub, and Skeleton

Core Java Modifiers

Differences between Java Collections

Java Collections Framework

Event Driven Programming

Event Driven Programming

Event Driven Programming

Exceptions (2)

Java exception handling

Top 5 Core Java Interview Questions

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

```

1 final class SerializeExample implements Serializable
2     transient File f;
3     public Ser() throws FileNotFoundException {
4         f = new File("c:\\temp\\filename");
5     }
6 }

```

Non memory objects like sockets, file handles, etc cannot be serialized, hence mark them as “transient”.

Note: **@Transient** annotation suggests that the object should not be persisted in Hibernate.

Q6. What value will the following method return?

```

1
2 public static int getSomeNumber( ){
3     try{
4         return 2;
5     } finally {
6         return 1;
7     }
8 }
9

```

A6. 1 is returned because ‘finally’ has the right to override any exception/returned value by the try..catch block. It is a bad practice to return from a finally block as it can suppress any exceptions thrown from a try..catch block. For example, the following code will not throw an exception.

```

1
2 public static int getSomeNumber( ){
3     try{
4         throw new RuntimeException( );
5     } finally {
6         return 12;
7     }
8 }
9

```

Q7. What can prevent execution of a code in a finally block?

A7. a) An end-less loop.

```

1
2 public static void main(String[ ] args) {
3     try {
4         System.out.println("This line is printed .....

```

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 MBea

☐ 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

```

5 //endless loop
6 while(true){
7 //...
8 }
9 }
10 finally{
11 System.out.println("Finally block is reached.")
12 }
13 }
14

```

b) System.exit(1) statement.

```

1
2 public class Temp {
3
4     public static void main(String[] args) {
5         try {
6             System.out.println("This line is printed before exit");
7             System.exit(1);
8         }
9         finally{
10            System.out.println("Finally block is executed");
11        }
12    }
13 }
14

```

c) Thread death or turning off the power to CPU.

d) An exception arising in a finally block itself.

e) Process p = Runtime.getRuntime().exec("");

If using Java 7 or later editions, use AutoCloseable statements within the try block.

Popular Posts

♦ 11 Spring boot interview questions & answers

857 views

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

825 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

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

311 views

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

301 views

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

292 views

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

286 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.



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

01: ♦ 15 Ice breake

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

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

◀ 02: ♥♦ 10 Java String class interview questions & answers

♦ 30+ FAQ Java Object Oriented Programming (i.e. OOP) interview

Q&As ▶

Posted in Reserved Key Words

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

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

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 Ja

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

	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 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

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 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 PHP
- Unix Interview Q&A
- ♥ 14 Unix interview questions
- ♥ Hidden Unix, C, sed and awk to v
- Shell script interview questions
- Unix history commands
- Unix remoting in
- Unix Sed commands
- Free Java Interview Questions
- Java Integration
- Java Beginner Interview Questions
- 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 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.