

Industrial strength Java/JEE Career Companion to open more doors

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

[Home](#) › [Interview](#) › [Pressed for time? Java/JEE Interview FAQs](#) › [FAQ JEE Job Interview Q&A Essentials](#) › ♦ Q29-Q53: Top 50+ JEE Interview Questions & Answers

♦ Q29-Q53: Top 50+ JEE Interview Questions & Answers

Posted on [February 24, 2015](#) by [Arulkumaran Kumaraswamipillai](#)

0
Like
Share

Tweet

0
G+1
Share

This JEE interview questions and answers cover some of the FAQs. This extends the other Java interview questions and answers on **enterprise Java development**

[EE Java interview Questions and Answers Q1 – Q28](#) and core Java development [50+ Core Java interview questions & answers](#)

Q29. What are the different ways can you wire up your dependencies using Spring?

A29. 3 different ways. You can combine all 3 ways.

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

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

- ✚ [Ice Breaker Interview](#)
- ✚ [Core Java Interview C](#)
- ✚ [JEE Interview Q&A \(3](#)
- ✚ [Pressed for time? Jav](#)
- ✚ [Job Interview Ice B](#)
- ✚ [FAQ Core Java Jot](#)
- ✚ [FAQ JEE Job Inter](#)
- ✚ [♦ 12 FAQ JDBC](#)
- ✚ [♦ 16 FAQ JMS ir](#)
- ✚ [♦ 8 Java EE \(ake](#)
- ✚ [♦ 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](#)
- ✚ [Java Application Ar](#)

- Using an XML based [application context file as demonstrated here in STEP 4](#) .
- Using annotations `@Resource`, `@Component`, `@repository` and `@Service`, and `@Autowired`. See [steps 2 and 3 in this Spring tutorial](#).
- Using the Java config using the annotations `@Configuration` and `@Bean`. This is demonstrated via [wiring up Spring JMS with Websphere MQ Listener \(Receiver or Subscriber\) in step2](#).

Q30-Q33. Can you describe the following bean concepts

- Spring managed beans life cycles.
- Bootstrapping the initial Spring bean.
- What is a bean factory?
- How would you create an application context from a web application?

A30-A33 The answers are clearly explained in this [13+ Spring FAQ interview questions and answers](#).

Q34. Can you list some real life scenarios where you will use Spring AOP?

A34.

1. You can use the Method AOP interceptors for dead lock retry and profiling your application. Occasionally, you get deadlocks in databases and your database management systems resolves deadlocks by aborting a transaction. These transactions can be retried with the help of Spring AOP method interceptors. Example 1: [Spring AOP with AspectJ for profiling Java applications](#), Example 2: [Q12 demonstrating deadlock retry with Spring and AspectJ](#).
2. Another area is service retry service in general where any failed remote service calls can be retried at a certain interval say every 10 seconds for 3 times.
3. for auditing purposes. [Logging and auditing in Java with Spring AOP tutorial](#).
4. Spring itself uses aspects for things like transaction management, security, and caching.

[+ Hibernate Job Inter](#)
[+ Spring Job Interview](#)
[+ Java Key Area Ess](#)
[+ OOP & FP Essential](#)
[+ Code Quality Job In](#)
[+ SQL, XML, UML, JSC](#)
[+ Hadoop & BigData Int](#)
[+ Java Architecture Inte](#)
[+ Scala Interview Q&As](#)
[+ Spring, Hibernate, & I](#)
[+ Testing & Profiling/Sa](#)
[+ Other Interview Q&A I](#)
[+ Free Java Interview](#)

16 Technical Key Areas

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

[+ Best Practice \(6\)](#)
[+ Coding \(26\)](#)
[+ Concurrency \(6\)](#)
[+ Design Concepts \(7\)](#)
[+ Design Patterns \(11\)](#)
[+ Exception Handling \(3\)](#)
[+ Java Debugging \(21\)](#)
[+ Judging Experience In](#)
[+ Low Latency \(7\)](#)
[+ Memory Management](#)
[+ Performance \(13\)](#)
[+ QoS \(8\)](#)
[+ Scalability \(4\)](#)
[+ SDLC \(6\)](#)
[+ Security \(13\)](#)
[+ Transaction Managen](#)

80+ step by step Java Tutorials

If the interview you are attending require Spring knowledge & experience, then go through the following link [17 Spring FAQ interview Questions & Answers](#).

Q35. What is the link between EntityManagerFactory/EntityManager and SessionFactory/Session?

A35. EntityManagerFactory and **EntityManager** are **JPA** (java Persistence API). **SessionFactory** and **Session** are **hibernate**-specific. Prefer EntityManagerFactory and EntityManager. The EntityManager invokes the hibernate session under the hood. And if you need some specific features that are not available in the EntityManager, you can obtain the session by calling

```
1 Session session = (Session) entityManager.getDele
```

Q36. Explain hibernate object states? Explain hibernate objects life cycle?

A36. Persistent objects and collections are short lived single threaded objects, which store the persistent state. These objects synchronize their state with the database depending on your flush strategy (i.e. auto-flush where as soon as setXXX() method is called or an item is removed from a Set, List etc or define your own synchronization points with session.flush(), transaction.commit() calls). If you remove an item from a persistent collection like a Set, it will be removed from the database either immediately or when flush() or commit() is called depending on your flush strategy. They are Plain Old Java Objects (POJOs) and are currently associated with a session. As soon as the associated session is closed, **persistent objects become detached objects** and are free to be used directly as data transfer objects in any application layers like business layer, presentation layer, etc.

[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 & Hlbernat](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

100+ Java pre-interview coding tests

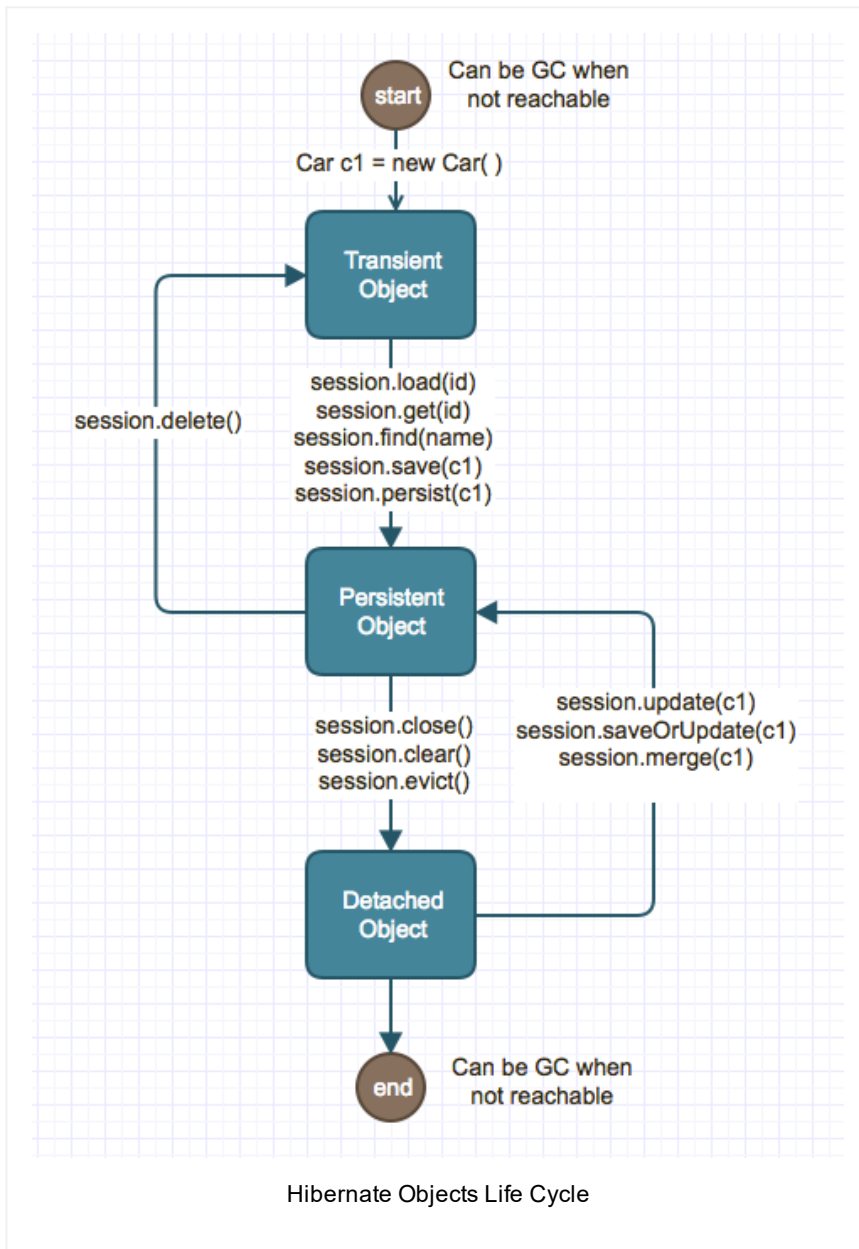
[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [♦ Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

How good are your?

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

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



Note: In JPA 2.0, you use an **EntityManager** instead of a Session. So, you will use `entityManager.persist(entity)` to create(..), `entityManager.merge(entity)` to edit(..), `entityManager.remove(this.entityManager.merge(entity))` to remove(..), and `entityManager.find(entityClass, primaryKey)` to find(..),

Detached objects and collections are instances of persistent objects that were associated with a session but currently not associated with a session. These objects can be freely used as Data Transfer Objects without having any impact on your database. Detached objects can be later on attached to another session by calling methods like

`session.update()`, `session.saveOrUpdate()` etc. and become persistent objects.

Transient objects and collections are instances of persistent objects that were never associated with a session. These objects can be freely used as Data Transfer Objects without having any impact on your database. Transient objects become persistent objects when associated to a session by calling methods like `session.save()`, `session.persist()` etc.

Note: The states of transient and detached objects cannot be synchronized with the database as they are not managed by Hibernate.

Q37. What are the general steps involved in creating Hibernate related class?

A37. The general steps involved in creating Hibernate related classes involve the following steps:

#1. Define the domain (aka entity) objects like Employee, Address, etc to represent relevant tables in the underlying database with the appropriate annotations or using the *.hbm.xml mapping files.

#2. Define the Repository (aka DAO — Data Access Objects) interfaces and implementations classes that use the domain objects and the hibernate session to perform data base CRUD (Create, Read, Update and Delete) operations the hibernate way.

#3. Define the service interfaces and the classes that make use of one or more repositories (aka DAOs) in a transactional context. A transaction manager will be used to coordinate transactions (i.e. commit or rollback) between a number of repositories.

#4. Finally, use an IoC container like Spring framework to wire up the Hibernate classes like SessionFactory, Session, transaction manager, etc and the user defined repositories, and the service classes. A number of interceptors can be

wired up as well for deadlock retry, logging, auditing, etc using Spring.

#5. Favor using JPA and CrudRepository from Spring.

If the interview you are attending require Hibernate knowledge & experience, then go through the following link [30+ Hibernate interview questions & answers](#). If you have time, go through the [other hibernate interview questions and answers](#) .

Q38-Q40 on Java Persistence API (**JPA**) are answered with diagrams at [JPA interview questions and answers](#)

Q38. What is a JPA? What are its key components?

Q39. What is the difference between hibernate.cfg.xml and persistence.xml?

Q40. What is an EntityManagerFactory and a Persistence unit? What is an EntityManager? What is an Entity?

Q41-Q45 on Java DataBase Connectivity (**JDBC**) are answered with diagrams at [12 JDBC interview questions and answers](#)

Q41. What are JDBC Statements? What are different types of statements? How can you create them?

Q42. What is the difference between statements and prepared statements? Which one would you favor and why?

Q43. What design pattern does JDBC use?

Q44. How will you bootstrap the JDBC driver?

Q45. Why lookup via JNDI for DataSources and other resources like JMS Queues, etc?

Q46-Q48 on Java Naming and Directory Interface (**JNDI**) and Lightweight Directory Access Protocol (**LDAP**) are answered with diagrams at [JNDI and LDAP interview questions and answers](#)

Q46. What is JNDI? And what are the typical uses within a JEE application?

Q47. What resources can you look up via a JNDI tree?

Q48. Why use LDAP when you can do the same with relational database (RDBMS)?

Q49-Q50 on Java Messaging Service (**JMS**) are answered with diagrams at [16 JMS interview questions & answers](#)

Q49. What is the difference between a “Queue” and a “Topic”? or What types of messaging paradigms are provided by JMS? or what is the difference between “point-to-point” and “publish-subscribe” paradigms relating to messaging?

Q50. What are the components of the JMS architecture?

Q51-Q53 on Java Server Faces (**JSF**) are answered with diagrams at [JSF interview questions & answers](#). Not all organizations use JSF. The emerging trend is to use JavaScript based frameworks like AngularJS, Backbone, Ember, etc. So, check your job spec before brushing up on JSF.

Q51. What are the 6 key phases of JSF?

Q52. What is a viewstate in JSF?

Q53. In your experience, what are some of the JSF pitfalls to watch out for?

Popular Posts

♦ [11 Spring boot interview questions & answers](#)

825 views

♦ [Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers](#)

767 views

[18 Java scenarios based interview Questions and Answers](#)

400 views

001A: ♦ [7+ Java integration styles & patterns interview questions & answers](#)

388 views

01b: ♦ [13 Spring basics Q8 – Q13 interview questions & answers](#)

295 views

♦ [7 Java debugging interview questions & answers](#)

293 views

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

285 views

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

279 views

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

240 views

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

201 views

Bio

Latest Posts



Arulkumaran Kumaraswamipillai



Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ ♦ Q01-Q28: Top 50+ EE Java interview questions & answers

02: ♦ Java 8 Streams, lambdas, intermediate vs terminal ops, and lazy

loading with simple examples ▶

Posted in FAQ JEE Job Interview Q&A Essentials, member-paid, Top 50+ FAQ

JEE Interview Q&A

Tags: JEE FAQs, TopX

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.