A Guide to Java Interview Questions

Have a planned interview in the coming days? This guide having 100+ **Java interview questions** will help you in revising most asked core java interview questions for experienced developers and asked in small startups to big corporate.

Start from these basic core java questions. I have written these tutorials specifically for helping you in answering some of the difficult questions you may face. I have tried to put as much reasoning as possible behind each answer. Still, it is not possible to include every possible question in this guide, so I have included some good resources at the end of this interview guide.

1. Core Java Interview Questions

1.1. Core Java Interview Questions Series

Generally, each interviewer will start with core Java concepts before jumping onto more advanced topics. The reason is simple that he wants to ease you and make you feel comfortable.

These questions may range from simple object-oriented principles to most used Java classes such as `String` or

Log4j Tutorial
Regex Tutorial

`HashMap`. I am saying them simple questions because they test your foundation, your basic knowledge, and your curiosity to go deeper into details. Try to answer all such questions. They are low hanging fruits which you should not miss.

This series is divided into 3 parts. These questions will help you in quickly revising most asked core java questions in a very short span of time. The best way to start with preparing for your next interview.

- Core Java Interview Questions – Part 1
- Core Java Interview Questions – Part 2
- Core Java Interview Questions – Part 3

1.2. Object Initialization Best Practices

In Java, **object initialization** is considered a heavy process and you are expected to know how each newly created object is affecting the memory and application performance. A simple example is Java wrapper classes which looks very easy from outside, just like **primitives**, but they are not as easy as they look. Learn how Java helps you with internal caching of objects inside wrapper classes such as `Double`, `Long` or `Integer`.

1.3. How HashMap Works in Java

`HashMap` is probably most discussed and controversial topic, if you are appearing in any junior or mid-level interview. You can face any interview question related to **`HashMap`**, if you know how hashmap works internally? This post will help you in answering some good questions like –

- How HashMap store key-value pairs?
- How HashMap resolve conflicts?
- How hashCode() and equals() method are used in HashMap?
- Impact of random/fixed hashCode() value for key?
- Using HashMap in multi-threaded environment?

1.4. Design a Good Key for HashMap

So you know now how `HashMap` works? Now learn about designing a good key for `HashMap`. This is a good way to test if you correctly understood HashMap's internal working. This will help you in answering questions like –

- Why `String` is a good key for HashMap?

- How you will design a class to be used as key?
- Will you override hashCode() method in Key class? What will be impacts?
- Write the syntax for a class which can be HashMap key object?

1.5. Questions on ConcurrentHashMap

HashMap is not thread-safe. We can use **HashTable** in concurrent applications, but it impacts the application performance. So we have `ConcurrentHashMap`. It is the concurrent version of `HashMap` which gives the same performance as HashMap and it is also thread-safe at the same time.

You are expected to know **ConcurrentHashMap** more depth. There are plenty of concepts in this single class alone. This is also another good topic to learn while preparing for your next interview.

1.6. Java Collections Interview Questions

Should I recommend you to prepare **Collection framework** and all of it's main classes? I guess you are already intelligent enough.

Any interview (junior and mid-level) will be incomplete if there are no questions related to the Collections framework. It truly tests your programming skills and your knowledge of core Java APIs. Questions may be as simple as Collections hierarchy to difficult ones such as Queue and Stacks. Here is a list of such commonly asked collections interview questions –

- Explain Collections hierarchy?
- Difference between **Set** and **List**?
- Difference between **Vector** and **ArrayList**?
- Difference between HashMap and HashTable?

- Difference between **Iterator** and **ListIterator**?
- Why Map interface does not extend Collection interface?
- How to convert an array of String to ArrayList?
- How to reverse the list?
- How HashSet store elements?
- Can a null element added to a **TreeSet** or **HashSet**?
- What are **IdentityHashMap** and **WeakHashMap**?
- When to use HashMap or TreeMap?
- How to make a collection read only?
- How to make a collection thread safe?
- What is difference between **fail-fast** and **fail-safe**?
- What is **Comparable** and **Comparator** interface?
- What are Collections and **Arrays** class?
- What is **Queue** and **Stack**? List their differences?

1.7. What is polymorphism in Java?

In simple words, **polymorphism** is the ability by which, we can create functions or reference variables which behaves differently in different programmatic context. Polymorphism is one of the major building blocks of object-oriented programming along with inheritance, abstraction, and encapsulation.

Learn the concept in more detail with examples. This is just so much important.

1.8. What is abstraction in Java?

In the previous question, you learned polymorphism. Now it's time to expand your knowledge by understanding **abstraction** as well. A very complicated topic for any Java interview.

1.9. Abstraction vs. Encapsulation?

Knowing the difference between abstraction and **encapsulation** is the key to understand both the concept in deep. You cannot learn either of both in isolation. They walk along in Java, so they must be understood collectively.

In this post, I have explained encapsulation as well as differentiated it with abstraction. A must read java interview question before moving forward.

1.10. Difference between Interfaces and Abstract Classes?

There has been very clear separation **abstract classes** and **interfaces** in Java since the language was born. But a lot has changed since Java 8 release came in market. Its one of core concept was functional interfaces.

Functional interfaces completely changed the way we look at both basic building blocks of Java language. You cannot skip this question if your resume says you work on Java 8. In the linked tutorial, I will show you the correct scenarios which will help you in cracking some complex interview questions as well as case studies.

1.11. Enum interview questions

Enum has been core building block for a long time. They can be seen in most popular Java libraries. They help you in managing constants in a more object-oriented manner. They look very easy but they hide lots of complexity if you dig deep enough. Some enum questions may be –

- Difference between enum vs. Enum class?
- Can enum be using with String?
- Can we extend enum?
- Write syntax of enum?
- How to implement reverse-lookup in enum?
- What is EnumMap and EnumSet?

1.12. Java Serialization and Serializable Interface

If you are preparing for Java interview with a Telecom company or any such domain who uses **serialization** in their application flows, then you will highly benefit from this tutorial. A very good list of do's and dont's with serialization in java. Possible questions may include –

- What is **serialVersionUID**?
- What is **readObject** and **writeObject**?
- ow you will serialize and deserialize a class?
- How you will make changes to a class so that serialization should not break?
- Can we serialize static fields?

1.13. Java Main Method

Ever wondered why `main()` is public, static and void? It's not a very frequently asked interview question in Java interviews but still I will recommend to read this post to answer these questions:

- Java main method syntax?
- Why main method is public?
- Why main method is static?
- Why main method is void?
- What happens internally when you invoke main method?

1.14. Java Object Clone

Object **cloning** in Java is not an easy concept. I myself took a long time to understand cloning in java. It really seems simple; use the `Cloneable` interface and override `clone()` method. But wait; there is much more to tell and ask in an interview. e.g.

- How clone() method works?
- What is shallow copy in Java?
- What are copy constructors?
- What is deep copy in Java?
- Different ways to create deep copy of an object?

1.15. What is CountDownLatch?

Since Java 5, `java.uti.concurrent` package has lots of useful but complex classes to work on concurrent applications. `CountDownLatch` is one of those classes which are highly asked in any Java interview big corporates. In this tutorial, `CountDownLatch` is explained with example and concepts around it.

1.16. Why Strings are Immutable?

This question is a very popular interview question at the beginner level. Basically, interviewer test your knowledge around String class, **string pool**, memory areas and object creation.

I wrote this post separately because the concept is so much important. In fact, **immutability** is itself a very important concept in java. Feel the tip of the iceberg.

1.17. How to Make a Java class immutable?

An immutable class is one whose state can not be changed once created. There are certain guidelines to create a class immutable in Java and you must know them to answer this question correctly.

Be aware that immutability is important in many design aspects and is a recommended design pattern by all Java

gurus. Learn to make a java class immutable, how it benefits the application design and be prepared to encounter more software design interview questions on it.

2. Java Interview Questions on Concurrency

2.1. What is Thread Safety?

Defining **thread safety** is surprisingly tricky. At the heart of any reasonable definition of thread safety is the concept of correctness. So, before understanding the thread-safety we should understand first, this “correctness”.

In this must-read Java tutorial, clear your doubts and be ready to answer some popular interview questions. e.g.

- What is correctness in thread safety?
- Give an example of thread-safe class?
- How you will design a thread safe Java class?
- Are immutable classes thread safe?

2.2. Object level locking vs. class level locking

At the heart of **concurrency**, there lie the concepts of object locking. Locking happens at instance level as well as class level.

- Object level locking is mechanism when you want to synchronize a non-static method or non-static code block such that only one thread will be able to execute the code block on given instance of the class. This should always be done to make instance level data thread safe.
- Class level locking prevents multiple threads to enter in a synchronized block in any of all available instances on runtime. This means if in runtime there are 100 instances of `DemoClass`, then only one thread will be able to execute `demoMethod()` in any one of the instances at a time, and all other instances will be locked for other threads. This should always be done to make static data thread safe.

Understand the whole concept in detail in this tutorial.

2.3. Difference between “implements Runnable” and “extends Thread”?

This is also a very popular interview question. If your role demands to create designs for concurrent applications then

you must know the correct answer to this question. The answer to this question decides which construct you choose between both.

Also, it will help you in answering basic questions such as –

- Difference between Thread and Runnable?
- Write Java code to create thread with Runnable interface?
- Which method should be preferred between both?

2.4. Compare and Swap [CAS] Algorithm

This question is targeted towards mid-level or senior developers. This requires a deep understanding of other concurrent concepts before answering this question. So It is a good way to test deep knowledge in Java concurrency.

- What is optimistic and pessimistic locking?
- What is compare and swap algorithm?
- What is an atomic operation?
- How AtomicInteger and AtomicLong works?

2.5. What is Fork/Join framework?

This is not a new concept but is now used in multiple ways since the release of Java 8. Fork-Join breaks the task at hand into mini-tasks until the mini-task is simple enough that it can be solved without further breakups. It's like a divide-and-conquer algorithm. One important concept to note in this framework is that ideally no worker thread is idle. They implement a work-stealing algorithm in that idle workers steal the work from those workers who are busy.

Learn this sweet and very efficient algorithm to better prepare for your next interview.

2.6. What is ThreadPoolExecutor?

In concurrent Java application, creating a thread is an expensive operation. And if you start creating new thread instance everytime to execute a task, application performance will degrade surely. ThreadPoolExecutor solves this problem.

ThreadPoolExecutor separates the task creation and its execution. With ThreadPoolExecutor, you only have to implement the Runnable objects and send them to the executor. It is responsible for their execution, instantiation, and running with necessary threads.

Read how ThreadPoolExecutor solves various problems and how it is used with BlockingQueue.

2.7. Java executor framework tutorial and best practices

You will learn the executors in the previous link, but there are certain things you need to know for using these executors effectively in terms of performance.

2.8. How to write a deadlock and resolve in Java

It can come in form of a puzzle. Better be ready for it. The interviewer may test your concurrency knowledge and your deep understanding on `wait()` and `notify()` method calls.

Be ready with one **deadlock** source-code example in your finger-tips. You will need it.

> Above given questions are must read before appearing any Java interview. Still, they do not provide complete coverage. You need to know more and more to perform better in interviews. I will suggest you to read more topics in the following sections in given order.

3. Java Interview Questions for Experienced Developers

3.1. Best Practices Guides

A list of highly recommended best practices. They will open up your mind to think in different directions. In short, they will sharpen your weapon for your next java interview.

3.2. Checkout some Puzzles

Solve these **puzzles** and play around them. You never know what will come in your way on a bad day.

3.3. Brush up Design Patterns

Aiming for a senior position in your next java interview, you MUST know these design patterns to handle some complex application design issues, which usually are discussed in all interviews nowadays.

3.4. Random browsing on Core Java Topics

Still got strength to read more. Browse more java interview questions here.

4. Spring Interview Questions

4.1. Spring Core Interview Questions

I have tried to collect some top spring core interview questions which you face into your next technical interview e.g.

- What is Inversion of Control (IoC) and Dependency Injection (DI)?
- Difference between BeanFactory and ApplicationContext?
- What is Spring Java-Based Configuration?
- Explain Spring Bean lifecycle?
- What are different Spring Bean Scopes?
- Are Singleton beans thread safe in Spring Framework?
- Explain different modes of bean autowiring?
- Explain @Qualifier annotation with example?
- Difference between constructor injection and setter injection?
- Name some of the design patterns used in Spring Framework?

4.2. Spring AOP Interview Questions

Spring **AOP** (Aspect Oriented Programming) compliments OOPs in the sense that it also provides modularity. In OOPs, key unit is Objects, but in AOP key unit is aspects or cross-cutting concerns such as logging and security. AOP provides the way to dynamically add the cross-cutting concern before, after or around the actual logic using simple pluggable configurations

Go through these most asked AOP interview questions-

- Difference between concern and cross-cutting concern?
- What are the available AOP implementations?
- What are the different advice types in spring AOP?
- What is Spring AOP Proxy?
- What is **Joint point** and **Point cut**?
- What is aspect weaving?

4.3. Spring MVC Interview Questions

These Spring MVC interview questions and answers have been written to help you prepare for the interviews and quickly revise the concepts in general. I will strongly suggest you to go deeper into each concept if you have extra time. In general, you should be able to answer these questions-

- What is MVC Architecture?
- What is DispatcherServlet and ContextLoaderListener?
- How to use Java based configuration?
- How can we use Spring to create Restful Web Service returning JSON response?
- Difference between <context:annotation-config> vs <context:component-scan>?
- Difference between @Component, @Controller, @Repository & @Service annotations?
- How does Spring MVC provide validation support?
- What is Spring MVC Interceptor and how to use it?
- How to handle exceptions in Spring MVC Framework?
- How to achieve localization in Spring MVC applications?

5. Test Your Knowledge

5.1. Real Java interview questions asked for Oracle Enterprise Manager Project

So far you have been learning all different concepts in Java which can come in front of you in form of interview questions. It's time to see whether you are prepared or not. Go through some real questions asked from Sreenath Ravva, in his interview with Oracle Corporation.

- Can you just start telling about your self and your project?
- What is abstraction and encapsulation in java ?
- Method Overloading rules?
- Widening and narrowing in java?
- Can I have only try block in code?
- Threads : producer and consumer problem?
- Why wait(), notify() and notifyAll() are defined in Object class?
- Can we override wait() or notify() methods?
- Difference between wait(), sleep() and yield()?
- Explain about join() method in thread class?

- Have you faced out of memory error? If yes how you fixed ? Tell different scenarios why it comes?
- Database connection leakage?
- Write a program to swap two numbers with out using third variable?
- Write a program to sort an array and remove duplicates?
- Write a program on Singleton?
- Write a program to merge two arrays?
- What is the use of final and finally keywords?
- Can I declare class as static or private?
- Why you want to change the company?

5.2. Java interview questions for mid-level developers

Nikhil has 6 years of experience as java/j2ee developer and he was planning to change the company. I suggested to him these intermediate level interview questions.

He was successful in his interview. You can take benefit from this as well.

6. Recommended Books

6.1. Java Puzzlers

Every programming language has its quirks. This book reveals the oddities of the Java programming language through programming puzzles.

6.2. Java Concurrency in Practice

Java Concurrency in Practice provides you with the concepts and techniques needed to write safe and scalable Java programs for today's–and tomorrow's–systems.

6.3. Effective Java by Joshua Bloch

Effective Java, Second Edition, brings together seventy-eight indispensable programmer's rules of thumb: working, best-practice solutions for the programming challenges you encounter every day.

7. Other online resources

Apart from these Java interview questions for freshers to experienced programmers, find time to read these articles

as well. They also contain a wealth of knowledge.

1. Java Flow Control Interview Questions (+ Answers) – Baelbung
2. Basic Java Interview Questions And Answers – Intellipaat
3. Top 75 Java Interview Questions – Edureka
4. More Java Interview Questions – JournalDev
5. Top 100 Java Interview Questions with Answers – Guru99
6. Commonly Asked Java Programming Interview Questions – Geeksforgeeks
7. 201 Core Java Interview Questions – Javatpoint
8. 100+ Core Java Interview Questions – Beginnersbook

Happy Learning !!

Feedback, Discussion and Comments

pravin

February 22, 2019

Hello Sir,
is this ques are relevant to 1 year of experience person please reply?

Reply

Sachin Sridhar

January 11, 2019

Awesome Sir.

Reply

katherine

December 8, 2018

Really this very usefull interview questions on Java you have given. I appreciate your work. Thanks for sharing the information. This is very useful information for the beginners.

Reply

Sharad

November 5, 2018

Thanks for comprehensive list of questions for Java Interview preparation.

Reply

shrija

September 25, 2018

Well, your blog is quite interesting and helpful. I just loved reading your blog. Thank you so much for java interview qestions. This would help many people around who are facing interview problems. Thank you for sharing this. it helped me.

Reply

Vishal Sinha

June 20, 2018

Hello Lokesh,

Great set of questions. I like that its not about syntax in these questions but many of these questions helps to find out the conceptual knowledge and good grasp over language and programming principles.

It helps me prepare and also gives me handy set of questions for evaluating someone else.

Regards
Vishal Sinha

Reply

arun singh

May 6, 2018

thanks sir

Reply

Shubhank

March 29, 2018

thanks for these information. Nice place for quick review.

Reply

Sunil

March 25, 2018

thx man for knowledge sharing.

Reply

pranit

February 5, 2018

Thanks a lot for sharing the top java interview question here, i have found your article very good and useful as i have an interview as java developer and was looking for some java questions to prepare for. Thanks a lot for sharing1

Reply

Ask Questions & Share Feedback

Your email address will not be published. Required fields are marked *

Comment

*Want to Post Code Snippets or XML content? Please use [java] ... [/java] tags otherwise code may not appear partially or even fully. e.g.

```
[java]
public static void main (String[] args) {
...
}
[/java]
```

Name *

Email *

Website

POST COMMENT

Meta Links

Advertise
Contact Us
Privacy policy
About Me