

# Top 10 Tough Core Java Interview Questions Answers for Programmers

---

 [java67.com/2012/09/top-10-tough-core-java-interview-questions-answers.html](http://java67.com/2012/09/top-10-tough-core-java-interview-questions-answers.html)

## **tough core Java interviews questions and answers**

What is tough core java interview question ? Why do people look for tough Java questions before going for an interview? well, I don't think I need to answer these tough questions because it's pretty natural to prepare for tough questions even if you are not expecting tough questions from core Java. If you are prepared for tough and tricky Java interview question than you feel more confident and answer other Java interview question with confidence. On the other hand, if you are not prepared for tough and tricky core Java questions than seeing them on Java interview or written test may surprise you. But the definition of tough core Java questions is not universal, same Java question which is easy for one programmer might be tough for other Java programmer. That's why it's best to prepare your own list of tough interview questions before appearing on any Java job interview. In this article, I am going to share you with my **Top 10 tough core Java interview questions and answers**, which may help you in Java interview.

## **10 tough Java interview question and answer**

---

Here is my list of *10 tough or tricky Java interview questions*. These questions are mostly from Core Java and I have not included J2EE questions. As I said you may know answers of these tough Java question or you may not even find it tough enough to challenge your Java knowledge but once upon a time these were asked in various Java interview and many programmers including my friends and colleagues find them tough to answer.

### **Why wait and notify is declared in Object class instead of Thread ?**

Another tough java question, how can you answer this question if you are not designed Java programming language. anyway, some common sense and deep knowledge of Java programming help to answer such tough core java interview question. See this blog post to learn Why to wait and notify is declared in Object class and not in Thread .

### **Why is multiple inheritances not supported in Java ?**

I found this *core Java question* really tough to answer because your answer may not satisfy Interviewer, in most cases Interviewer is looking for specific points and if you can bring them, they would be happy. The key to answering this kind of tough question in Java is to prepare topic well to accommodate any follow-ups. See Why multiple inheritances are not supported in Java for an answer of this tough Java question.

### **Why does Java not support operator overloading ?**

One more similar category of tough Java question. C++ supports operator overloading than why not Java? this is the argument Interviewer will give to you and sometimes even say that + operator is overloaded in Java for String concatenation, Don't be fooled by such arguments. See Why to support operator overloading is not supported in Java for detailed answer of this tricky Java question.

### Why is String immutable in Java?

My favorite Java interview question, this is tough, tricky but same time very useful as well. Some interviewer also asks this question as Why String is final in Java. look at this post for some points which make sense on [Why String is final or immutable in Java](#)

### Why char array is preferred to store password than String in Java?

Another [tricky Java question](#) which is based on String and believe me there is only few Java programmer which can answer this question correctly. This is a real *tough core Java interview question* and again the solid knowledge of String is required to answer this. see [Why char array is better than String for storing the password in Java](#) to find out the answer to this tough Java question.

### How to create a thread-safe singleton in Java using double-checked locking?

This Java question is also asked as What is a [thread-safe](#) singleton and how to do you write it. Well, Singleton created with double checked locking before Java 5 was a broker and it's possible to have multiple instances of Singleton if multiple threads try to create an instance of Singleton at the same time. from Java 5 it's easy to [create thread-safe Singleton using Enum](#). but if interviewer persists with double checked locking then you have to write that code for them. remember to use a volatile variable. See [10 Java singleton interview question](#) for more details on this topic.

50 x 50

### Write Java program to create a deadlock in Java and fix it ?

One of the classical but *tough core Java interview question* and you are likely to fail if you have not involved in the coding of multi-threaded concurrent Java application. See [how to create and prevent deadlock in Java](#) for complete answer of this tough core Java interview question

### What happens if your Serializable class contains a member which is not serializable?

#### How do you fix it?

Any attempt to Serialize that class will fail with NotSerializableException, but this can be easily solved by making that variable [transient](#) for static in Java. See [Top 10 Serialization interview question answers in Java](#) for more details.

### Why wait and notify called from synchronized method in Java?

Another tough core Java question for the wait and notify. They are called from [synchronized method or synchronized block](#) because wait and modify need monitor on Object on which wait or notify-get called. See [Why to wait and notify require synchronized context](#) for a complete answer to this *tough and tricky Java multi-threading question*.

### Can you override static method in Java? if I create the same method in the subclass is it compile time error?

No, you can not override [static method in Java](#) but it's not a compile-time error to declare an exactly same method in a subclass, That is called method hiding in Java. See [Can you override static method in Java](#) for a complete answer to this tough Java interview question.

1  
1  
1

1  
1  
1