# 104 Java Debugging interview questions to hire top engineers

## Questions

1. Imagine your program is a car, and it's not moving. How do you find out why it's stuck?

2. What's the simplest way to see what a variable is holding at a specific point in your code, like peeking inside a box?

3. If your program is doing something you didn't expect, how do you slow it down to watch it step-by-step?

4. What's a 'breakpoint,' and how does it help you catch errors in your Java code, like setting a trap for a bug?

5. How can you tell if a specific part of your code is even being run, like checking if a room is being used?

6. Your code throws an 'exception.' What does that even MEAN, and how do you find where it happened?

7. What's the difference between stepping 'into' a function and stepping 'over' it while debugging?

8. If you change a variable's value while debugging, does that permanently change your code?

9. What's the deal with 'watch expressions' in debuggers? Can you give a simple use case?

10. Can you debug code on a remote server? What tools or techniques are needed?

11. What are some common mistakes that lead to NullPointerExceptions, and how can you catch them early?

12. How do you debug multithreaded Java applications? What special challenges arise?

13. Explain the concept of a 'core dump' and how it aids in debugging production issues.

14. Let's say your application is running very slow. How would you start investigating performance bottlenecks using debugging tools?

15. Your program compiles fine, but at runtime it is not working as expected. Where do you even start?

16. What are the advantages and disadvantages of using a debugger versus simply adding print statements to your code?

17. How can you use conditional breakpoints to stop your program only when a variable has a specific value?

18. Describe a time when you used debugging to solve a particularly tricky problem. What tools did you use, and what was your approach?

19. What are some best practices for writing code that is easy to debug?

20. How can you use a debugger to inspect the contents of a collection (like a List or Map) at runtime?

21. Explain how you might debug a unit test that is failing.

22. What are some common debugging keyboard shortcuts in your favorite Java IDE (like IntelliJ or Eclipse)? How do they speed up the debugging process?

23. How would you go about debugging a memory leak in a Java application?

24. Describe a situation where using a logging framework (like Log4j or SLF4J) would be more effective than using a debugger.

25. How do you set conditional breakpoints in your IDE, and why would you use them?

26. Explain the difference between 'step into', 'step over', and 'step out' in a debugger.

27. What is a watch expression in debugging, and how can it help you?

28. How can you debug a multithreaded Java application?

29. What are some common issues that can make debugging multithreaded code difficult?

30. Describe how you would use remote debugging to diagnose a problem in a deployed Java application.

31. What are some security considerations when using remote debugging?

32. How can you use logging frameworks (like Log4j or SLF4j) to aid in debugging?

33. What are some best practices for writing effective log messages?

34. Explain how to use memory analysis tools (like a heap dump analyzer) to diagnose memory leaks or excessive memory usage.

35. What are some common causes of memory leaks in Java applications?

36. How can you use profiling tools (like JProfiler or VisualVM) to identify performance bottlenecks in your code?

37. What are some common performance issues that can be identified using profiling tools?

38. Describe how you would debug a NullPointerException. What steps would you take to find the root cause?

39. How would you debug a situation where your application is throwing an unexpected exception?

40. How do you debug code that involves reflection?

41. Explain how you would debug code that uses lambda expressions or streams.

42. What are some challenges associated with debugging asynchronous code, and how can you overcome them?

43. How can you use assertions to help debug your code?

44. What are the limitations of using assertions for debugging?

45. Describe a situation where you used a debugger to solve a complex problem in a Java application.

46. Explain what a core dump is and how it can be used for debugging crashes.

47. How do you analyze thread dumps to diagnose deadlocks or performance issues?

48. What strategies do you use to debug integration tests?

49. How do you approach debugging code written by someone else, especially if it's poorly documented?

50. What are some common mistakes that developers make when debugging, and how can you avoid them?

51. Explain how you would debug a situation where a database query is performing slowly. What tools would you use?

52. How can you debug memory leaks in a Java application without using a profiler?

53. Explain the process of debugging a multi-threaded application where threads are deadlocking.

54. What strategies can be used to debug performance bottlenecks in Java applications running in production?

55. How would you debug a Java application that is consistently throwing OutOfMemoryError?

56. What are some techniques to debug race conditions in concurrent Java programs?

57. Describe the steps you would take to debug a Java application using remote debugging.

58. How do you debug classloading issues or NoClassDefFoundError in Java?

59. Explain how you would debug a situation where a Java application is consuming excessive CPU resources.

60. What tools and techniques can you use to debug issues related to garbage collection in Java?

61. How would you approach debugging a problem where a Java application's response time is unpredictable?

62. Explain how to debug issues in Java applications that involve native libraries (JNI).

63. What are some advanced techniques for debugging asynchronous code in Java?

64. How would you debug a Java application that is interacting with a database and experiencing slow query performance?

65. Describe how to debug issues in a Java application that arise only under heavy load or stress.

66. How do you debug issues related to serialization and deserialization in Java?

67. Explain how you would debug a Java application experiencing network connectivity problems.

68. What techniques can you use to debug code generated by annotation processors in Java?

69. How would you approach debugging a Java application running inside a Docker container?

70. Describe the process of debugging a Java application deployed on a cloud platform like AWS or Azure.

71. How do you debug memory corruption issues in Java when using native libraries?

72. Explain how to debug issues related to bytecode manipulation libraries like ASM or Javassist.

73. What strategies can you use to debug complex regular expressions in Java?

74. How would you debug an issue where a Java application is not properly handling character encoding?

75. Describe how to debug security vulnerabilities in Java code, such as SQL injection or cross-site scripting.

76. How would you debug a Java application that's failing due to library version conflicts?

77. Explain your approach to debugging issues caused by reflection in Java.

78. What advanced techniques do you use to debug issues in reactive programming with frameworks like Reactor or RxJava?

79. How would you debug a Java application that is behaving differently in different environments (dev, staging, prod)?

80. Explain how to debug issues with caching mechanisms in Java applications, such as using Ehcache or Redis.

81. What is the process of debugging performance issues when using Java streams and parallel processing?

82. How would you debug a memory leak in a long-running Java application, and what tools would you use?

83. Explain how you would approach debugging a multi-threaded application where race conditions are suspected.

84. Describe a scenario where traditional debugging methods are ineffective, and how you would overcome this challenge.

85. How would you debug a performance bottleneck in a Java application without using a profiler?

86. What strategies do you use to debug issues in production environments without impacting users?

87. How would you debug a complex Spring application with numerous dependencies and layers?

88. Describe your experience debugging issues related to Java garbage collection. What tools and techniques do you employ?

89. How do you approach debugging intermittent or non-deterministic bugs in Java?

90. Explain how you would debug an application that is crashing with an OutOfMemoryError.

91. What debugging techniques do you use when dealing with asynchronous code, such as CompletableFuture or RxJava?

92. Imagine you're debugging code you didn't write. What's your process for understanding and fixing the bug?

93. How would you debug a Java application that is integrated with a message queue like Kafka or RabbitMQ?

94. How do you typically debug a deadlock situation in a Java application?

95. Describe your experience debugging complex SQL queries generated by a Java application. How do you optimize slow queries?

96. Explain how you would approach debugging a high CPU utilization issue in a Java application.

97. How do you debug issues related to class loading in Java, such as ClassNotFoundException or NoClassDefFoundError?

98. What are your preferred methods for debugging remote Java applications, and what are the challenges involved?

99. How do you debug a security vulnerability discovered in a Java application?

100. Let's say you're debugging a complex algorithm, how do you verify the correctness of the logic?

101. How do you debug issues related to serialization and deserialization in Java?