
35+ Java Interview Questions for 2026

A Complete Hiring & Evaluation Toolkit for Recruiters, Hiring Managers & Candidates

How to Use This Guide

This playbook helps:

- Recruiters run structured interviews
- Hiring managers evaluate effectively
- Candidates prepare comprehensively

SECTION 1: BEGINNER JAVA INTERVIEW QUESTIONS (0–2 YEARS)

1. What is Java?

Java is an object-oriented, platform-independent programming language that runs on the JVM (Java Virtual Machine). It follows the principle of "Write Once, Run Anywhere."

Strong Answer: Mentions JVM, bytecode, portability, and use cases like web and enterprise apps.

2. Difference between JDK, JRE, and JVM

- JVM: Executes Java bytecode
- JRE: Provides runtime environment
- JDK: Development kit including JRE + tools

3. What are OOP principles?

- Encapsulation: Wrapping data and methods
- Inheritance: Reusing code from parent class
- Polymorphism: Same method, different behavior
- Abstraction: Hiding implementation details

4. What is method overloading vs overriding?

- Overloading: Same method name, different parameters
- Overriding: Redefining method in subclass

5. What are access modifiers?

- public: Accessible everywhere
- private: Within class only
- protected: Within package + subclasses
- default: Within package

6. What is a constructor?

A special method used to initialize objects. It has the same name as the class and no return type.

7. What is inheritance?

Mechanism where one class acquires properties of another using extends keyword.

8. What is abstraction?

Hiding internal implementation using abstract classes or interfaces.

9. What is polymorphism?

Ability to take multiple forms (compile-time and runtime polymorphism).

10. What is String immutability?

Strings cannot be changed after creation, improving security and performance.

SECTION 2: INTERMEDIATE JAVA QUESTIONS (2–5 YEARS)

11. Difference between ArrayList and LinkedList

- ArrayList: Dynamic array, faster access
- LinkedList: Node-based, faster insertion/deletion

12. What is HashMap and how does it work?

Stores key-value pairs using hashing. Uses buckets and handles collisions via chaining or tree structures.

13. HashMap vs HashTable

- HashMap: Not synchronized
- HashTable: Thread-safe but slower

14. What is exception handling?

Mechanism to handle runtime errors using try-catch-finally blocks.

Best Practice: Catch specific exceptions, not generic Exception.

15. Checked vs Unchecked exceptions

- Checked: Compile-time (IOException)
- Unchecked: Runtime (NullPointerException)

16. What is multithreading?

Running multiple threads concurrently to improve performance.

17. What is synchronization?

Controls thread access to shared resources to avoid data inconsistency.

18. String vs StringBuilder vs StringBuffer

- String: Immutable
- StringBuilder: Mutable, not thread-safe
- StringBuffer: Mutable, thread-safe

19. equals() vs ==

- == compares references
- equals() compares values

20. What is garbage collection?

Automatic memory management that removes unused objects.

SECTION 3: ADVANCED JAVA QUESTIONS (5+ YEARS)

21. JVM Architecture

Includes Class Loader, Runtime Data Areas (Heap, Stack), Execution Engine, and Garbage Collector.

22. What are design patterns?

Reusable solutions to common problems.

Examples:

- Singleton
- Factory
- Observer

23. What is Spring Framework?

A framework for building enterprise Java applications with features like dependency injection and MVC architecture.

24. What is dependency injection?

Providing dependencies externally instead of creating them inside the class.

25. What is Hibernate?

ORM framework that maps Java objects to database tables.

26. What is microservices architecture?

Breaking applications into small, independent services.

27. What is REST API?

Web service that uses HTTP methods like GET, POST, PUT, DELETE.

28. What is concurrency?

Handling multiple tasks at the same time efficiently.

29. What is thread pool?

Collection of reusable threads to improve performance.

30. What is CompletableFuture?

Used for asynchronous programming in Java.

SECTION 4: CODING QUESTIONS (WITH APPROACH)

31. Reverse a String

Approach: Use loop or StringBuilder reverse()

32. Check Palindrome

Approach: Compare string with reversed string

33. Find duplicates in array

Approach: Use HashSet

34. Fibonacci Series

Approach: Iterative or recursion

35. Two-sum problem

Approach: Use HashMap for $O(n)$

SECTION 5: REAL-WORLD SCENARIOS

36. API latency issue

Profiling, caching, DB optimization

37. Memory leak issue

Analyze heap dump, fix object references

38. System scaling problem

Load balancing, microservices, caching

SECTION 6: EVALUATION CHECKLIST

Skills	5/5	3/5	1/5
Core Java	Strong	Basic	Weak
Problem Solving	Optimized	Works	Poor
Debugging	Structured	Trial	None
System Design	Scalable	Limited	None
Communication	Clear	Avg	Poor

SECTION 7: HIRING SCORECARD

Skills	Scorecard (__ / 10)
Technical Skills	
Problem Solving	
System Design	
Communication	
Culture Fit	

Need help hiring top Java developers faster? [Talk to our experts today.](#)