

# Java Interview Questions

## 1. Basic Java

1. What are the main features of Java?
2. Explain the difference between **JDK, JRE, and JVM**.
3. What is the difference between **==** and **.equals()** in Java?
4. What are **primitive data types** in Java?
5. What is the purpose of the final keyword?
6. What is the difference between **static** and **non-static** methods?
7. Explain the concept of **method overloading** and **method overriding**.
8. What is the difference between **String, StringBuilder, and StringBuffer**?
9. What is the super keyword in Java?
10. Explain the use of the this keyword.

## 2. Object-Oriented Programming (OOPs) Concepts

11. What are the four **OOPs** principles in Java?
12. What is **abstraction**, and how is it implemented in Java?
13. What is **encapsulation**, and why is it important?
14. What is the difference between an **abstract class** and an **interface**?
15. Explain **polymorphism** with an example.
16. What is a **constructor**, and how is it different from a method?
17. Can we override a **static method** in Java?
18. What is **inheritance**, and what are its types?
19. What is the purpose of the instanceof operator?
20. What is a **default method** in an interface

## 3. Exception Handling

21. What is **exception handling** in Java?
22. Explain the difference between **checked and unchecked exceptions**.
23. What is the purpose of the try-catch-finally block?
24. How does the throw and throws keyword work?
25. Can we have multiple catch blocks for a single try block?
26. What is the difference between **final, finally, and finalize**?
27. What happens if an exception is not handled in Java?

# Java Interview Questions

- 28. Can we write a try block without a catch block?
- 29. What is a **custom exception** in Java?
- 30. What is the use of the throws keyword in method declaration?

## 4. Java Collections Framework (JCF)

- 31. What is the Java **Collections Framework**?
- 32. What is the difference between **ArrayList** and **LinkedList**?
- 33. What is the difference between **HashMap** and **HashTable**?
- 34. What is the difference between **Set** and **List**?
- 35. What is the difference between **TreeSet**, **HashSet**, and **LinkedHashSet**?
- 36. What is a **ConcurrentHashMap**?
- 37. Explain the difference between **Iterator** and **ListIterator**.
- 38. What is the difference between **Comparable** and **Comparator**?
- 39. What is the **fail-fast** and **fail-safe** iterator?
- 40. How does **HashMap** work internally?

## 5. Multithreading & Concurrency

- 41. What is a **thread** in Java?
- 42. How do you create a thread in Java?
- 43. What is the difference between **Runnable** and **Thread**?
- 44. What is **synchronization**, and why is it needed?
- 45. What is the difference between **synchronized method** and **synchronized block**?
- 46. What is the difference between **volatile** and **synchronized**?
- 47. What is the difference between **notify()** and **notifyAll()**?
- 48. What is a **Thread Pool**?
- 49. What is the difference between **Callable** and **Runnable**?
- 50. What is the purpose of the **join()** method in threads?

## 6. Java 8 Features

- 51. What are the new features introduced in **Java 8**?
- 52. What is a **lambda expression** in Java?
- 53. What are **functional interfaces** in Java 8?
- 54. What is **Stream API**, and how does it work?

# Java Interview Questions

- 55. What is the purpose of the **default and static methods** in interfaces?
- 56. What is the difference between **map()** and **flatMap()** in Stream API?
- 57. What is **Optional**, and why is it used?
- 58. What is **method reference** in Java 8?
- 59. What is **Collectors.toList()** in Stream API?
- 60. What are some common **built-in functional interfaces** in Java?

## 7. Advanced Java & Miscellaneous

- 61. What is the difference between **JDBC and Hibernate**?
- 62. How does **Garbage Collection (GC)** work in Java?
- 63. What is the purpose of the **WeakHashMap**?
- 64. Explain **Spring Framework** in Java.
- 65. What is **Dependency Injection (DI)** in Spring?
- 66. What is **JPA (Java Persistence API)**?
- 67. What is the difference between **SOAP and REST APIs**?
- 68. What is **Microservices Architecture**?
- 69. How does **Docker** work with Java applications?
- 70. What is the difference between **Servlet and JSP**?