

# Java Interview – 67 Questions & Answers (Simple 3-Line Format)

## 1) What is JDK, JRE, JVM?

JDK is the kit with tools + JRE. JRE has libraries + JVM. JVM runs bytecode and executes the program.

## 2) What is Java bytecode?

Bytecode is the .class file after compilation. It is platform-independent and runs on JVM.

## 3) Features of Java?

Java is simple, object-oriented, platform-independent, secure, robust, multithreaded, and portable.

## 4) Is JVM platform independent?

No, JVM is platform dependent, but bytecode is platform independent, so Java runs anywhere.

## 5) What is class and object?

Class is a blueprint or design. Object is an instance of the class that represents real-world entities.

## 6) What is implicit and explicit casting?

Implicit casting is automatic widening conversion. Explicit casting is manual narrowing conversion.

## 7) Difference between while and do..while.

while checks condition before loop, may not run at all. do..while runs at least once before checking.

## 8) What is encapsulation?

Encapsulation means wrapping variables and methods together, and hiding details using access modifiers.

## 9) Object oriented concepts.

OOPs main pillars: Encapsulation, Inheritance, Polymorphism, and Abstraction for real-world modeling.

## 10) What is multiple inheritance? Why not in Java?

Multiple inheritance means extending 2 classes. Java doesn't support it to avoid ambiguity problems.

## 11) Scope of local, instance and static variables?

Local variables exist inside methods, instance with objects, static shared across all objects.

## 12) Example with Emp e=new Employee(), Emp e1=null?

e is a reference pointing to Employee object. e1 is a reference but pointing to null.

## 13) What is overloading and overriding?

Overloading means same method name with different parameters. Overriding means redefining method in subclass.

## 14) Can you overload final method?

Yes, final methods can be overloaded because overloading depends on method signature, not inheritance.

## 15) What is final modifier in Java?

final keyword makes variable constant, method non-overridable, and class non-inheritable.

## 16) Valid modifiers for a Java class?

Classes can be public, default, final, or abstract in Java.

## 17) Difference between protected and default?

protected gives access in same package and subclasses. default gives access only within package.

**18) If method is protected in superclass, what modifiers in subclass?**

It can be protected or public, but not private or default (must not reduce visibility).

**19) Similarities of abstract and interface?**

Both cannot be instantiated and both provide abstraction in Java.

**20) Difference between abstract and interface?**

Abstract class can have both concrete and abstract methods. Interface has only abstract (till Java 7).

**21) Default access modifier for method/variable in interface?**

Methods are public abstract by default, variables are public static final by default.

**22) What is interface inheritance?**

Interface inheritance means one interface extends another. Used to achieve full abstraction.

**23) Abstract method and concrete method?**

Abstract method has no body, must be implemented. Concrete method has full definition.

**24) How to choose between abstract and interface?**

Use abstract class when objects are related, use interface to define contracts across different classes.

**25) What is a constructor? Mandatory to define?**

Constructor initializes objects. Not mandatory, compiler provides default constructor if none is given.

**26) If parameterized constructor defined, need default constructor?**

Yes, if you want to use no-arg constructor, you must explicitly define it.

**27) What is constructor overloading?**

Defining multiple constructors with different parameters in same class is constructor overloading.

**28) What is constructor chaining?**

Calling one constructor from another within same class or superclass is chaining.

**29) What is this and super keyword?**

this refers to current object, super refers to immediate parent class object or methods.

**30) Difference between String and StringBuffer?**

String is immutable, StringBuffer is mutable and thread-safe for modifications.

**31) Difference between StringBuffer and StringBuilder?**

Both are mutable. StringBuffer is synchronized (thread-safe), StringBuilder is faster but not synchronized.

**32) What is naming collision in Java?**

When two classes have same names, we use package names to resolve collision.

**33) Use of packages in Java?**

Packages group related classes and interfaces together for reusability and easy maintenance.

**34) What is Object class?**

Object class is the parent of all classes. Methods like toString(), equals(), hashCode() can be overridden.

**35) Why override equals and hashCode?**

To compare object contents and ensure correct behavior in collections like HashMap.

**36) Superclass in exception hierarchy?**

Throwable is the superclass, with Error and Exception as subclasses.

**37) Explain Error and Exception.**

Error is unrecoverable (OutOfMemory), Exception is recoverable during program execution.

**38) What is an Exception?**

Exception is an abnormal condition that interrupts program flow and can be handled.

**39) Checked vs Unchecked exception?**

Checked exceptions must be handled at compile time, unchecked occur at runtime only.

**40) What is the use of finally block?**

finally block always executes to release resources like files or database connections.

**41) Difference between throw and throws?**

throw is used to explicitly throw exception, throws is used to declare exceptions in method.

**42) What is exception propagation?**

It means exception is passed from one method to another until handled or program ends.

**43) Order of multiple catch blocks?**

Always catch child exceptions first, then parent, otherwise compile error occurs.

**44) What is nested try block?**

A try block inside another try block is nested try, used for handling complex exceptions.

**45) What is custom exception?**

User-defined exception class that extends Exception for specific error handling.

**46) Explain collection framework.**

Collection framework provides classes like List, Set, Map to store and manipulate groups of objects.

**47) What is autoboxing and unboxing?**

Autoboxing converts primitive to wrapper, unboxing converts wrapper to primitive automatically.

**48) Difference between List and Set?**

List allows duplicates and ordered, Set does not allow duplicates and unordered.

**49) Difference between HashSet and TreeSet?**

HashSet is unordered and fast, TreeSet is sorted but slower.

**50) What is iterator?**

Iterator is used to traverse elements of a collection one by one.

**51) Explain generics in Java?**

Generics allow type safety in collections and classes, reducing runtime errors.

**52) Difference between HashMap and TreeMap?**

HashMap is unordered and faster, TreeMap is sorted in ascending key order.

**53) Difference between Iterator and ListIterator?**

Iterator can traverse forward only. ListIterator can traverse both directions and modify list.

**54) What is ClassCastException?**

Thrown when object is cast to incompatible type at runtime.

**55) What is NullPointerException?**

Thrown when you try to access or call methods on a null object reference.

**56) Comparator vs Comparable?**

Comparable defines natural order (compareTo), Comparator defines custom order (compare).

**57) How to sort a list in Java?**

Use Collections.sort() or list.stream().sorted() for sorting.

**58) Explain JDBC.**

JDBC is Java API to connect and interact with relational databases using SQL.

**59) What is a driver file?**

Driver file is software that connects Java with database (like MySQL connector).

**60) Difference between Statement and PreparedStatement?**

Statement executes static queries. PreparedStatement is precompiled, faster and prevents SQL injection.

**61) What is SQL injection?**

SQL injection is attack by inserting malicious SQL. Prevented using PreparedStatement.

**62) Advantage of new Date & Time API?**

New API is immutable, thread-safe and easier than old Date/Calendar classes.

**63) Difference between byte stream and character stream?**

Byte streams handle raw data (8-bit), character streams handle text (16-bit Unicode).

**64) What is stream in Java?**

Stream represents a sequence of objects for functional-style operations in Java 8.

**65) When to use buffered streams?**

Buffered streams are used for faster I/O by reducing direct access to disk.

**66) What is Lambda expression?**

Lambda is a short block of code that implements functional interface in one line.

**67) Explain Stream API in Java 8.**

Stream API processes collections with operations like filter, map, reduce in functional style.