

give me the core java posible interview question and answar

Certainly! Here are some of the most **commonly asked Core Java interview questions with concise answers**. These questions cover a broad range of topics, from basics to OOPs, collections, and exceptions.

1. What is Java?

Java is an object-oriented, high-level, platform-independent programming language developed by Sun Microsystems. Java code runs on the Java Virtual Machine (JVM) which makes it platform-independent.

2. What are the features of Java?

- Simple
- Object-Oriented
- Platform-Independent ("Write once, run anywhere")
- Secure
- Robust
- Architecture Neutral
- Portable
- Multi-threaded
- High Performance (with JIT)
- Distributed

3. What is the difference between JDK, JRE, and JVM?

- **JDK (Java Development Kit):** Contains tools required for Java development (compiler, debugger, etc.) + JRE.
- **JRE (Java Runtime Environment):** Contains JVM + libraries for running Java applications.
- **JVM (Java Virtual Machine):** Executes the compiled Java bytecode.

4. What is the main method signature?

```
public static void main(String[] args)
```

It is the entry point of any Java program.

5. What is an object and a class?

- **Class:** Blueprint for objects. It defines properties (fields) and behaviors (methods).
- **Object:** An instance of a class.

6. What is inheritance?

Inheritance is an OOP concept where one class (child/subclass) acquires the properties and behaviors of another (parent/superclass).

7. What is polymorphism?

Polymorphism allows one interface to be used for different data types. It can be *compile-time* (method overloading) or *runtime* (method overriding).

8. What is method overloading and overriding?

- **Overloading:** Creating multiple methods in the same class with the same name but different arguments.
- **Overriding:** Redefining a parent class method in a subclass with the same signature.

9. What is encapsulation?

Encapsulation is wrapping data (variables) and methods into a single unit (class) and restricting direct access using access modifiers (like `private`).

10. What is abstraction?

Abstraction means hiding internal implementation details and exposing only the essential functionalities via abstract classes or interfaces.

11. Difference between abstract class and interface?

Abstract Class	Interface
Can have abstract and concrete methods	Only abstract methods (default methods also from Java 8)
Can have constructors and fields	Cannot have constructors (till Java 8)
Supports single inheritance	Can be implemented by multiple classes (multiple inheritance)

12. What is a constructor?

A constructor is a special method with the same name as the class, called automatically when the object is created. It does not have a return type.

13. What is the default value of an object reference declared as an instance variable?

`null`

14. What is the use of the `final` keyword?

Used to mark classes as non-inheritable, methods as non-overridable, and variables as constants.

15. What is the difference between `==` and `.equals()`?

- `==` checks for reference equality (if two references point to the same object).
- `.equals()` checks for value/content equality.

16. What is a package in Java?

A package is a namespace that organizes related classes and interfaces. Example: `java.util`, `java.lang`.

17. What are access modifiers?

- **private** – accessible only in the class
- **default** (no modifier) – accessible within the package
- **protected** – accessible within package and by subclass
- **public** – accessible everywhere

18. What is exception handling?

Mechanism to handle runtime errors. It uses `try`, `catch`, `finally`, `throw`, and `throws`.

19. Checked vs. Unchecked Exceptions?

- **Checked:** Checked at compile time (e.g., `IOException`).
- **Unchecked:** Checked at runtime (e.g., `NullPointerException`).

20. What is a thread? How do you create threads in Java?

A thread is a lightweight process for multitasking.

- By extending the `Thread` class
- By implementing the `Runnable` interface

21. What is synchronization?

Synchronization is a mechanism to control access to a shared resource by multiple threads to prevent data inconsistency.

22. What is a String? Is it mutable?

String is an object in Java representing a sequence of characters. Strings are **immutable**.

23. Difference between StringBuffer and StringBuilder?

Both are mutable, but:

- **StringBuffer** is thread-safe (synchronized).
- **StringBuilder** is not thread-safe but faster.

24. What is the use of the `static` keyword?

It indicates that the member belongs to the class, not to any object. For example, static variables, methods, and blocks.

25. What is the use of the `super` keyword?

`super` refers to the parent class object. It is used to access parent class members and constructors.

26. What is garbage collection in Java?

Automatic process of deleting unused/null objects to free up memory.

27. Difference between ArrayList and LinkedList?

ArrayList	LinkedList
Fast random access	Slow random access
Slow insertion/deletion	Fast insertion/deletion
Backed by array	Backed by doubly linked list

28. What is the difference between List, Set, and Map?

- **List:** Ordered, allows duplicates (e.g., ArrayList).
- **Set:** Unordered, no duplicates (e.g., HashSet).
- **Map:** Key-value pairs, keys unique (e.g., HashMap).

29. What is autoboxing and unboxing?

- **Autoboxing:** Automatic conversion of primitive type to its wrapper class (int to Integer).
- **Unboxing:** Automatic conversion of wrapper class to primitive.

30. What is the difference between `throw` and `throws`?

- **throw:** Used to explicitly throw an exception.
- **throws:** Used in method signature to declare that a method may throw exceptions.

31. What is serialization?

Serialization is the process of converting an object into a byte stream to save/transmit it. Deserialization reverts it back.

32. Explain the lifecycle of an applet?

`init()`, `start()`, `paint()`, `stop()`, `destroy()`

33. How to prevent inheritance of a class?

Use the `final` keyword with the class.

34. Can you override a private or static method?

Private methods cannot be overridden. Static methods can be hidden (not overridden).

35. What are wrapper classes?

Classes that wrap primitive data types into objects (e.g., `Integer`, `Double`).

36. What is the default package in Java?

`java.lang`

37. Difference between `throw` and `throws`?

See Q#30.

38. What is method signature?

Method name + parameter list (type, number, and order of parameters).

39. What is the purpose of the `transient` keyword?

It prevents fields from being serialized.

40. What is the Java Collection Framework?

A set of classes and interfaces for storing and manipulating groups of data as a single unit (e.g., List, Set, Map).

If you need questions and answers on specific Java topics (e.g., threads, collections, OOPs separately), **mention that** and I will tailor the list for you!