

JAVA – TRICKY QUESTIONS & SAFE ANSWERS

JAVA – TRICKY QUESTIONS & SAFE ANSWERS

(All Companies | All Fresher Roles)

◆ JVM / MEMORY TRICKS

Q1. Is Java 100% object-oriented?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. Java is not 100% object-oriented because it uses primitive data types like int, char, etc.

Q2. Where are objects stored in Java?

✗ Wrong: Stack

✓ SAFE ANSWER:

Objects are stored in Heap memory. References are stored in Stack.

Q3. Does Java use pointers?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. Java does not allow direct pointer access. It uses references for safety.

Q4. Is garbage collection guaranteed?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. Garbage collection is automatic, but JVM decides when to run it.

◆ OOP CONCEPT TRICKS

Q5. Can we create object without constructor?

✗ Wrong: No

✓ SAFE ANSWER:

No. Constructor is always called. If not written, default constructor is provided by Java.

Q6. Can constructor be private?

✗ Wrong: No

✓ SAFE ANSWER:

Yes. Private constructors are used in Singleton design pattern.

Q7. Can we override static methods?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. Static methods belong to class, not object. They can be hidden, not overridden.

Q8. Can we overload main method?

✗ Wrong: No

✓ SAFE ANSWER:

JAVA – TRICKY QUESTIONS & SAFE ANSWERS

Yes. We can overload main method, but JVM calls only standard main method.

◆ ~~3~~ == vs equals() (VERY COMMON TRAP)

Q9. Difference between == and equals()?

✗ Wrong: Both same

✓ SAFE ANSWER:

== compares references. equals() compares content of objects.

Q10. Which one to use for String comparison?

✗ Wrong: ==

✓ SAFE ANSWER:

equals() should be used to compare String values.

◆ ~~4~~ STRING TRICKS

Q11. Are String objects mutable?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. String objects are immutable in Java.

Q12. Why String is immutable?

✗ Wrong: Just design

✓ SAFE ANSWER:

For security, thread safety, and performance (String pool).

Q13. Difference between String and StringBuilder?

✗ Wrong: Same

✓ SAFE ANSWER:

String is immutable. StringBuilder is mutable and faster.

◆ 5 EXCEPTION HANDLING TRICKS

Q14. What happens if exception is not handled?

✗ Wrong: Program continues

✓ SAFE ANSWER:

Program terminates abnormally.

Q15. Can we have try without catch?

✗ Wrong: No

✓ SAFE ANSWER:

Yes, if finally block is present.

Q16. Can finally block be skipped?

✗ Wrong: Never

✓ SAFE ANSWER:

Yes, in rare cases like `System.exit()`.

◆ 6 INHERITANCE & INTERFACE TRICKS

Q17. Can Java support multiple inheritance?

✗ Wrong: Yes

✓ SAFE ANSWER:

Java does not support multiple inheritance with classes, but supports it using interfaces.

Q18. Can interface have methods with body?

✗ Wrong: No

✓ SAFE ANSWER:

Yes. From Java 8, interfaces can have default and static methods.

Q19. Can interface have variables?

✗ Wrong: No

✓ SAFE ANSWER:

Yes. Interface variables are public, static, and final by default.

◆ 7 MULTITHREADING BASIC TRICKS (FRESHER SAFE)

Q20. What is thread?

✗ Wrong: Program

✓ SAFE ANSWER:

Thread is a lightweight process that executes tasks independently.

Q21. Difference between process and thread?

✗ Wrong: Same

✓ SAFE ANSWER:

Process is heavy and independent. Thread is lightweight and shares memory.

◆ 8 OUTPUT-BASED TRICK QUESTIONS

Q22.

```
String s1 = "Java";  
String s2 = "Java";  
System.out.println(s1 == s2);
```

✗ Wrong: False

✓ SAFE ANSWER:

True. Both refer to same String from String pool.

Q23.

```
String s1 = new String("Java");  
System.out.println(s1 == "Java");
```

✗ Wrong: True

✓ SAFE ANSWER:

JAVA – TRICKY QUESTIONS & SAFE ANSWERS

False. new creates different object.

◆ GENERAL JAVA TRAPS

Q24. Is Java faster than C?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. C is faster due to direct hardware access.

Q25. Is Java only used for backend?

✗ Wrong: Yes

✓ SAFE ANSWER:

No. Java is used in backend, desktop, mobile, and enterprise applications.

● GOLDEN SAFE LINE (Teach Students)

> "I may not remember exact syntax, but conceptually this works like this..."

This line protects students in any company interview.