https://www.youtube.com/DebugWithShubham

https://instagram.com/debugwithshubham

Cognizant Gen

https://www.linkedin.com/in/debugwithshubham/

https://topmate.io/debugwithshubham

https://web.telegram.org/k/#@debugwithshubham

DEBUG WITH SHUBHAM
TECH CHANNEL

cognizant

Cognizant GenC 2025

TOP- 20 JAVA

Interview Question With Answers

By- Shubham Maurya

debugwithshubham debugwithshubham debugwithshubham

Q1. What is Java? Why is it platform independent?

Answer:

Java is a high-level, object-oriented programming language.
Java platform independent hai kyun ki code bytecode me convert hota hai jo JVM pe run hota hai

One-liner:

Write once, run anywhere.

Q2. What is JVM, JRE, and JDK?

Answer:

JVM → Java Virtual Machine (runs bytecode)

JRE → JVM + libraries (to run Java programs)

JDK → JRE + development tools (to develop programs)

Q3. What is OOPs? Explain its pillars.

Answer:

OOPs organizes code using objects.

4 pillars:

-Encapsulation

-Abstraction

-Inheritance

-Polymorphism

Q4. What is Class and Object?

Answer:

-Class → Blueprint

-Object → Instance of class

Example:

```
class Car {}
Car c = new Car();
```

Q5. Difference between == and equals()

Answer:

== → compares reference
equals() → compares content/value

Q6. What is Constructor? Types?

Answer:

Constructor is a special method used to initialize objects.

Types:

- -Default constructor
- -Parameterized constructor

-Constructor name = class name.

Q7. What is Inheritance?

Answer:

Inheritance allows a class to acquire properties of another class using extends.

Benefits:

- -Code reuse
- -Maintainability

Q8. What is Polymorphism?

Answer:

Polymorphism means one method, multiple behaviors.

Types:

- -Compile-time (Method Overloading)
- -Runtime (Method Overriding)

Q9. Difference between Method Overloading and Overriding

Overloading	Overriding
Same class	Parent–Child class
Compile-time	Runtime
Different parameters	Same method signature

Q10. What is Abstraction?

Answer:

Abstraction hides internal details and shows only essential features.

Achieved using:

- -Abstract class
- -Interface

Q11. Interface vs Abstract Class

Abstract Class	Interface
Can have method body	Only abstract methods (Java 7)
Supports constructor	No constructor
Single inheritance	Multiple inheritance

Q12. What is Encapsulation?

Answer:

Encapsulation binds data and methods together and protects data using access modifiers.

Example:

Private variables + public getters/setters

Q13. What are access modifiers in Java?

Answer:

-public

-private

-protected

-default

Controls visibility of data.

Q14. What is Exception Handling?

Answer:

Used to handle runtime errors using:

try
catch
finally
throw
throws

Example:

```
try {
int a = 10/0;
} catch(Exception e) {
System.out.println("Error");
}
```

Q15. Checked vs Unchecked Exception

Checked	Unchecked
Compile-time	Runtime
IOException	NullPointerException

Q16. What is ArrayList vs LinkedList?

ArrayList	LinkedList
Fast access	Fast insertion/deletion
Uses array	Uses nodes

Q17. What is String, StringBuffer, StringBuilder?

String	StringBuffer.	StringBuilder
Immutable	Mutable.	Mutable.
Thread-safe (NO)	Thread-safe (YES)	Thread-safe (NO)

Q18. What is static keyword?

Answer:

Static members belong to class, not object.

Used for memory management.

Q19. What is final keyword?

Answer:

final variable → value cannot change

final method → cannot override

final class → cannot inherit

Q20. Why Java is preferred in enterprise applications?

Answer:

Because Java is secure, scalable, platform-independent, and has strong community support