

# JAVA

## OOPS – HARD LEVEL

1. Which of the following is true about **covariant return types** in Java?

- A) They allow overriding methods to return subtype objects
- B) They require same return type strictly
- C) They only apply to constructors
- D) They are only valid with interfaces

**Answer: A**

2. Which of these can achieve **multiple inheritance of behavior** in Java?

- A) Multiple abstract classes
- B) Multiple interfaces with default methods
- C) Multiple final classes
- D) Static methods only

**Answer: B**

3. Which OOP principle does the **Dependency Inversion Principle (DIP)** emphasize?

- A) High-level modules should not depend on low-level modules
- B) A class must have only one responsibility
- C) Derived classes must substitute base classes
- D) Objects should be open for extension but closed for modification

**Answer: A**

4. Which happens if two interfaces have default methods with the same signature and a class implements both?

- A) Compilation error unless overridden
- B) One is chosen arbitrarily
- C) JVM resolves automatically
- D) Both run in order of declaration

**Answer: A**

5. Which is **not** true about abstract classes?

- A) Can have constructors
- B) Can extend other abstract classes
- C) Can be final
- D) Can contain static methods

**Answer: C**

6. Which is true about **downcasting** in Java?

- A) Always safe if superclass reference points to a subclass object
- B) No runtime checks are performed
- C) Requires explicit casting
- D) Can be done without instanceof check safely

**Answer: C**

7. Which design pattern is an example of encapsulation?

- A) Singleton
- B) Factory
- C) Observer
- D) Composite

**Answer: A**

8. Which OOP feature is most related to the **Liskov Substitution Principle**?

- A) Inheritance
- B) Polymorphism
- C) Abstraction
- D) Encapsulation

**Answer: B**

9. Which principle ensures objects should be open for extension but closed for modification?

- A) Encapsulation
- B) Open/Closed Principle
- C) Dependency Inversion
- D) Interface Segregation

**Answer: B**

10. Which is the correct order of constructor execution in a multilevel inheritance chain?

- A) Child → Parent → Grandparent
- B) Parent → Child → Grandparent
- C) Grandparent → Parent → Child
- D) JVM decides randomly

**Answer: C**

---

11. Which is true about **method hiding** in Java?

- A) Happens with static methods
- B) Happens with private methods
- C) Same as overriding
- D) Allows polymorphism

**Answer: A**

12. Which principle states "Many client-specific interfaces are better than one general-purpose interface"?

- A) Liskov Substitution
- B) Interface Segregation
- C) Dependency Inversion
- D) Encapsulation

**Answer: B**

13. Which is true about overriding exceptions?

- A) Overriding method can throw broader checked exception
- B) Overriding method can throw narrower checked exception
- C) Overriding method must throw the same exception
- D) Exceptions are ignored in overriding

**Answer: B**

14. Which keyword ensures a method cannot be accessed outside its class hierarchy?

- A) final
- B) private
- C) abstract
- D) protected

**Answer: B**

15. Which of these allows creating **immutable classes** in Java?

- A) Declaring fields as final and private
- B) No setters
- C) Returning copies instead of references
- D) All of the above

**Answer: D**

16. Which design pattern provides a way to create objects without exposing instantiation logic?

- A) Singleton
- B) Factory
- C) Observer
- D) Adapter

**Answer: B**

17. Which Java OOP feature is broken by exposing internal mutable objects?

- A) Abstraction
- B) Encapsulation
- C) Polymorphism
- D) Inheritance

**Answer: B**

18. Which happens if a constructor is declared private?

- A) Class cannot be extended
- B) Class cannot be instantiated outside
- C) Class must be abstract
- D) JVM throws error

**Answer: B**

19. Which is true about **abstract methods**?

- A) They can be declared static
- B) They must be implemented in subclass
- C) They can have body
- D) They must be private

**Answer: B**

20. Which concept allows binding method calls to method bodies at runtime?

- A) Static binding
- B) Dynamic binding
- C) Compile-time polymorphism
- D) Operator overloading

**Answer: B**

---

21. Which is the correct statement about super() call in constructors?

- A) Must be the first statement if used
- B) Can be anywhere in constructor
- C) Optional but can follow other statements
- D) Cannot be used in constructors

**Answer: A**

22. Which type of class is created when you define an inner class inside a method?

- A) Local inner class
- B) Anonymous class
- C) Static nested class
- D) Lambda class

**Answer: A**

23. Which OOP principle is applied when composition is preferred over inheritance?

- A) Encapsulation
- B) Aggregation
- C) Loose coupling
- D) Polymorphism

**Answer: C**

24. Which keyword prevents overriding but allows inheritance?

- A) final
- B) abstract
- C) static
- D) protected

**Answer: A**

25. Which mechanism ensures that an interface can provide implementation without breaking old code?

- A) Multiple inheritance
- B) Default methods (Java 8+)
- C) Static methods
- D) Abstract classes

**Answer: B**

26. Which method is used by JVM for garbage collection hint?

- A) finalize()
- B) dispose()
- C) clean()
- D) gc()

**Answer: A**

27. Which of these is NOT a valid reason to use abstract classes?

- A) Provide partial implementation
- B) Avoid object creation
- C) Ensure subclasses must override
- D) Achieve multiple inheritance

**Answer: D**

28. Which is true about final classes?

- A) Cannot be inherited
- B) Can contain final methods
- C) Can have constructors
- D) All of the above

**Answer: D**

29. Which keyword allows accessing hidden fields of parent class?

- A) super
- B) this
- C) parent
- D) final

**Answer: A**

30. Which OOP concept is demonstrated when multiple classes implement the same interface differently?

- A) Polymorphism
- B) Encapsulation
- C) Inheritance
- D) Abstraction

**Answer: A**

---

31. Which of the following breaks encapsulation?

- A) Exposing mutable fields directly
- B) Providing only getters
- C) Making fields private
- D) Returning immutable copies

**Answer: A**

32. Which OOP concept is closely related to overriding equals() and hashCode() properly?

- A) Polymorphism
- B) Encapsulation
- C) Object equality contract
- D) Inheritance

**Answer: C**

33. Which of these can be used to implement Singleton pattern safely in multithreaded Java?

- A) Lazy initialization without sync
- B) Eager initialization
- C) Double-checked locking with volatile
- D) Both B and C

**Answer: D**

34. Which class loading mechanism is followed by JVM?

- A) Child-first delegation
- B) Parent-first delegation
- C) Random order
- D) Explicit user control only

**Answer: B**

35. Which OOP concept is used in **method overloading resolution**?

- A) Runtime polymorphism
- B) Compile-time polymorphism
- C) Dynamic binding
- D) Abstract binding

**Answer: B**

36. Which allows creating object-specific behavior at runtime in Java?

- A) Anonymous classes
- B) Static classes
- C) Final classes
- D) Abstract classes only

**Answer: A**

37. Which is true about interfaces with static methods?

- A) Inherited automatically by implementing class
- B) Must be overridden
- C) Cannot be overridden
- D) Must be private

**Answer: C**

38. Which type of class cannot be serialized by default?

- A) Abstract class
- B) Final class
- C) Inner class without static modifier
- D) Public class

**Answer: C**

39. Which OOP concept is broken if subclass violates parent's method contract?

- A) Polymorphism
- B) Liskov Substitution Principle
- C) Abstraction
- D) Encapsulation

**Answer: B**

40. Which of these is true about object cloning in Java?

- A) Requires implementing Cloneable interface
- B) clone() is defined in Object class
- C) Shallow copy is created by default
- D) All of the above

**Answer: D**

---

41. Which is true about **composition over inheritance**?

- A) It promotes stronger coupling
- B) It allows more flexibility
- C) It prevents polymorphism
- D) It requires abstract classes

**Answer: B**

42. Which OOP principle ensures classes depend on abstractions rather than concrete implementations?
- A) Dependency Inversion
  - B) Open/Closed Principle
  - C) Encapsulation
  - D) Composition

**Answer: A**

43. Which mechanism supports polymorphism in Java at bytecode level?
- A) Static binding
  - B) Virtual method table (vtable)
  - C) Reflection API
  - D) Method hiding

**Answer: B**

44. Which is true about abstract classes with no abstract methods?
- A) Not allowed
  - B) Allowed, prevents instantiation
  - C) Must have at least one abstract method
  - D) JVM rejects them

**Answer: B**

45. Which is the most restrictive access level for class members?
- A) public
  - B) protected
  - C) private
  - D) default

**Answer: C**

46. Which mechanism ensures type safety at runtime in Java?
- A) Reflection
  - B) Generics with type erasure
  - C) instanceof checks
  - D) Dynamic binding

**Answer: C**

47. Which keyword is used to define a constant in an interface?
- A) const
  - B) final static
  - C) define
  - D) abstract

**Answer: B**

48. Which design principle says "favor object composition over class inheritance"?
- A) Encapsulation principle
  - B) Composition principle
  - C) GoF OOP principle
  - D) None

**Answer: C**

49. Which OOP concept is violated if equals() is overridden but hashCode() is not?

- A) Encapsulation
- B) Object contract
- C) Abstraction
- D) Polymorphism

**Answer: B**

50. Which mechanism allows Java to support reflection?

- A) Metadata in bytecode
- B) vtable at runtime
- C) JVM optimization
- D) Garbage collector hooks

**Answer: A**