Java Developer Interview Questions and Answers

1. What is Java?
Java is a high-level, object-oriented programming language developed by Sun Microsystems. It is platform-independent due to the Java Virtual Machine (JVM).

2. What are the features of Java?
* Object-Oriented
* Platform Independent
* Secure
* Robust
* Multithreaded
* High Performance

3. What is the JVM?
The Java Virtual Machine (JVM) is an engine that provides a runtime environment to run Java bytecode.

4. What is JRE?
The Java Runtime Environment (JRE) provides libraries and JVM to run Java applications.

5. What is JDK?
The Java Development Kit (JDK) includes JRE and development tools like the compiler and debugger.

6. What is the difference between JDK, JRE, and JVM?
* JDK = JRE + Development tools
* JRE = JVM + Libraries
* JVM = Java Virtual Machine

7. What are Java's primitive data types?
byte, short, int, long, float, double, boolean, char

8. What is the difference between Array and ArrayList?
Arrays are fixed in size; ArrayLists are dynamic and only hold objects.

9. What is a constructor in Java?
A constructor is a special method used to initialize objects.

10. What is method overloading?
Method overloading allows multiple methods with the same name but different parameters.

11. What is method overriding?

Overriding means redefining a parent class's method in the subclass.

12. What is the difference between == and equals()?
`==` compares references; `.equals()` compares values.

13. What is encapsulation?
Encapsulation means hiding the internal state of an object and only exposing required parts via methods.

14. What is inheritance?
Inheritance allows a class to inherit properties and behavior from another class.

15. What is polymorphism? -
Polymorphism means one interface, many implementations. It can be method overloading or overriding.

16. What is an interface?
An interface is a reference type in Java, similar to a class, that contains method declarations but no implementation.

17. What is an abstract class?
An abstract class cannot be instantiated. It may contain abstract methods that must be implemented in derived classes.

18. What are wrapper classes?
They convert primitive types into objects (e.g., int → Integer).

19. What is autoboxing/unboxing?
Autoboxing: primitive → object
Unboxing: object → primitive

20. What is a package in Java?
A package is a namespace for organizing classes and interfaces.

21. What are access modifiers in Java?
* public
* private
* protected
* default (package-private)

22. What does the static keyword do?
Declares members that belong to the class, not instances.

23. What is the final keyword used for?

* Final variable: can’t be reassigned
* Final method: can’t be overridden
* Final class: can’t be extended

24. What is the this keyword?
Refers to the current object.

25. What is the super keyword?
Refers to the parent class’s members.

26. What is exception handling?
Exception handling manages runtime errors with try, catch, finally, throw, and throws.

27. Difference between checked and unchecked exceptions?
Checked: checked at compile-time
Unchecked: checked at runtime

28. What is the use of finally block?
It always executes whether an exception is thrown or not.

29. Difference between throw and throws?
* `throw`: used to throw an exception
* `throws`: used in method signature to declare exceptions

30. What is a try-with-resources block?
A try block that declares resources and closes them automatically.

31. What is the Collection Framework?
A set of interfaces and classes for storing and manipulating data.

32. Difference between List, Set, and Map?
* List: ordered, allows duplicates
* Set: unordered, no duplicates
* Map: key-value pairs

33. What is a HashMap?
A non-synchronized collection that stores key-value pairs.

34. What is a TreeMap?
A map that maintains keys in sorted (natural) order.

35. What is a LinkedHashMap?
Maintains insertion order.

36. What is a HashSet?
Implements Set, backed by a HashMap. No duplicate elements.

37. What is TreeSet?
Implements SortedSet and stores elements in ascending order.

38. What is an Iterator?
Used to iterate through collections one element at a time.

39. Difference between Iterator and ListIterator?
* Iterator: forward only
* ListIterator: forward and backward

40. What is ConcurrentHashMap?
A thread-safe variant of HashMap.

41. What is multithreading in Java?
Allows multiple threads to run concurrently for better performance

42. Ways to create threads in Java?
* Extend Thread class
* Implement Runnable interface
* Use ExecutorService

43. What is synchronization?
Control access of multiple threads to shared resources.

44. What is a volatile variable?
A variable whose value is always read from main memory, ensuring visibility across threads.

45. What is deadlock?
Two or more threads waiting indefinitely for each other's resources.

46. What are daemon threads?
Background threads that don't prevent JVM from exiting.

47. What is garbage collection?
Automatic memory management that reclaims memory occupied by unused objects.

48. What is finalize() method?
Called by garbage collector before object is removed. Deprecated in modern Java.

49. What are lambda expressions?
A concise way to represent anonymous functions (Java 8+).

50. What is the Stream API?
Provides functional-style operations for processing sequences of elements.

51. What is the difference between String, StringBuilder, and StringBuffer?
* `String`: Immutable
* `StringBuilder`: Mutable, not thread-safe
* `StringBuffer`: Mutable, thread-safe

52. What is immutability in Java?
An immutable object cannot be changed after it is created. `String` is a classic example.

53. How are Strings stored in memory?
Strings are stored in a string pool to optimize memory usage and performance.

54. What is String interning?
Interning is storing only one copy of each distinct string value in a common pool.

55. Difference between == and equals() for strings?
* `==`: reference comparison
* `equals()`: value/content comparison

56. What is an enum in Java?
A special class used to define a fixed set of constants.

57. What is the default value of an uninitialized boolean variable?
`false`

58. What is a transient variable?
A variable that is not serialized when an object is serialized.

59. What is a volatile variable?
A variable whose changes are always visible to all threads.

60. What is the difference between shallow and deep copy?
* Shallow copy: copies references
* Deep copy: copies actual objects recursively

61. What is reflection in Java?
An API that allows inspection and modification of classes, methods, and fields at runtime.

62. What is the Singleton design pattern?
A design pattern that ensures a class has only one instance and provides a global point of access to it.

63. How do you implement Singleton in Java?
By making the constructor private and providing a static method to return the instance.

64. What is a factory design pattern?
Creates objects without exposing the creation logic to the client.

65. What is the Observer pattern?
A pattern where an object (subject) notifies other objects (observers) when its state changes.

66. What are annotations in Java?
Metadata that provides information to the compiler or runtime tools (e.g., `@Override`, `@Deprecated`).

67. What is a functional interface?
An interface with a single abstract method, e.g., `Runnable`, `Callable`.

68. What is the default method in interface?
A method in an interface with a body, introduced in Java 8.

69. What are method references in Java 8?
A shorthand notation of a lambda expression to call a method.

70. What is the Optional class in Java 8?
A container object used to contain not-null objects and avoid `NullPointerException`.

71. What is the difference between HashMap and Hashtable?
* `HashMap` is not synchronized
* `Hashtable` is synchronized and thread-safe

72. What is the diamond operator?
Introduced in Java 7 to simplify generic instance creation: `List<String> list = new ArrayList<>();`

73. What is type erasure in generics?
The process by which generic type information is removed during compilation.

74. Can you overload static methods?
Yes, static methods can be overloaded.

75. Can you override static methods?
No, static methods cannot be overridden.

76. Can constructors be inherited?
No, constructors are not inherited.

77. Can an interface extend another interface?
Yes, an interface can extend multiple interfaces.

78. Can a class implement multiple interfaces?
Yes, Java supports multiple inheritance through interfaces.

79. What is the marker interface?
An interface with no methods, used to convey metadata (e.g., `Serializable`).

80. What is serialization?
The process of converting an object into a byte stream for storage or transmission.

81. What is deserialization?
The process of converting a byte stream back into an object.

82. What is the purpose of the transient keyword?
To prevent serialization of a variable.

83. What is the difference between composition and inheritance?
* Inheritance: "is-a" relationship
* Composition: "has-a" relationship

84. What is dependency injection?
A design pattern used to implement IoC, allowing object creation and dependency management externally.

85. What is inversion of control (IoC)?
A principle where control of objects or portions of a program is transferred to a container or framework.

86. What is the difference between compile-time and runtime polymorphism?
* Compile-time: method overloading
* Runtime: method overriding

87. What is the default value of local variables?
They must be initialized before use; they don’t have default values.

88. Can we override a private or static method?
No, private methods are not visible to subclasses; static methods belong to the class.

89. What is the difference between public, protected, and private?
* `public`: accessible everywhere
* `protected`: accessible within package and subclass

* `private`: accessible only in the same class

90. What is a no-arg constructor?
A constructor with no parameters.

91. What is an anonymous class?
A class without a name, defined and instantiated in a single statement.

92. What is method hiding in Java?
When a subclass defines a static method with the same name as a static method in the parent class.

93. What is a memory leak in Java?
A condition where objects are no longer needed but not garbage collected due to lingering references.

94. What is the purpose of the System.gc() method?
It requests garbage collection, but it's not guaranteed to run immediately.

95. What is the difference between break and continue?
* `break`: exits the loop
* `continue`: skips the current iteration

96. What are varargs in Java?
Allows a method to accept a variable number of arguments.

97. What is the purpose of the assert keyword?
To test assumptions during development using assertions.

98. Can we declare a constructor as final?
No, constructors cannot be final, static, or abstract.

99. What is the purpose of the synchronized keyword?
Used to control access to a method or block by multiple threads.

100. What is the difference between abstract class and interface?
* Abstract class: can have constructors and state
* Interface: only method signatures and static/final fields (before Java 8)

101. What are common exceptions in Java?
* NullPointerException
* ArrayIndexOutOfBoundsException
* ClassCastException
* ArithmeticException