TOP 100 JAVA INTERVIEW QUESTIONS AND ANSWERS

Basic Java Questions

1. **What is Java?**

   Answer: Java is a high-level, object-oriented programming language developed by Sun Microsystems, now owned by Oracle.

2. **What are the main features of Java?**

   Answer: Platform independence, object-oriented, multithreading, security, garbage collection, and high performance.

3. **What is JDK, JRE, and JVM?**

   Answer:

   - **JDK (Java Development Kit):** A software development kit for Java, including JRE and development tools.
   - **JRE (Java Runtime Environment):** Provides libraries and the JVM to run Java applications.
   - **JVM (Java Virtual Machine):** A virtual machine that runs Java bytecode.

4. **What is the difference between JDK and JRE?**

   Answer: JDK is a development kit with tools to create Java applications, while JRE is used to run Java programs.

5. **What is a class in Java?**

   Answer: A class is a blueprint for creating objects, defining properties (fields) and behaviors (methods).

Object-Oriented Programming (OOP) Concepts

6. **What is inheritance in Java?**

   Answer: Inheritance is a mechanism where a new class acquires the properties and behavior of an existing class.

7. **What is polymorphism in Java?**

   Answer: Polymorphism allows objects to take multiple forms, enabling method overloading and overriding.

8. **What is encapsulation in Java?**

   Answer: Encapsulation is the process of binding data (fields) and methods that operate on the data into a single unit (class).

9. **What is abstraction in Java?**

   Answer: Abstraction hides the complexity and shows only the essential features of an object.

10. **What is the difference between an abstract class and an interface?**

    Answer: An abstract class can have both abstract and non-abstract methods, while an interface only has abstract methods (prior to Java 8).

Java Data Types

11. **What are the primitive data types in Java?**

    Answer: byte, short, int, long, float, double, char, and boolean.

12. **What is the difference between primitive types and reference types in Java?**

    Answer: Primitive types store actual values, while reference types store references to objects.

13. **What is autoboxing and unboxing in Java?**

    Answer: Autoboxing is converting a primitive to its wrapper class, and unboxing is converting a wrapper class back to its primitive type.

14. **What is the difference between String and StringBuilder in Java?**

    Answer: String is immutable, while StringBuilder is mutable and used for manipulating strings.

15. **What is the final keyword in Java?**

    Answer: The final keyword can be applied to variables (constant), methods (cannot be overridden), and classes (cannot be inherited).

Control Flow

16. **What is the difference between == and equals() in Java?**

    Answer: == compares object references, while equals() compares the content of objects.

17. **What is the use of the break and continue statements?**

    Answer: break exits the loop, while continue skips the current iteration and moves to the next one.

18. **What are loops in Java?**

    Answer: Java supports three types of loops: for, while, and do-while.

19. **What is the purpose of the switch statement?**

    Answer: It allows multi-way branching based on the value of an expression.

20. **What is the difference between for loop and foreach loop?**

    Answer: for loop is used for iteration with an index, while foreach is used to iterate over collections or arrays.

Exception Handling

21. **What is exception handling in Java?**

    Answer: Exception handling is a mechanism to handle runtime errors using try, catch, and finally blocks.

22. **What is the difference between checked and unchecked exceptions?**

    Answer: Checked exceptions are exceptions that the compiler forces you to handle (e.g., IOException), while unchecked exceptions are runtime exceptions (e.g., NullPointerException).

23. **What is the try-catch block in Java?**

    Answer: The try block contains code that may throw exceptions, and the catch block handles them.

24. **What is the finally block in Java?**

    Answer: The finally block contains code that will always execute after the try and catch blocks, even if an exception occurs.

25. **What is the purpose of the throw keyword?**

    Answer: It is used to explicitly throw an exception.

Java Collections

26. **What is a collection in Java?**

    Answer: A collection is a framework that provides a set of interfaces and classes to store and manipulate groups of objects.

27. **What is the difference between ArrayList and LinkedList?**

    Answer: ArrayList uses a dynamic array to store elements, while LinkedList uses a doubly linked list.

28. **What is a HashMap in Java?**

    Answer: HashMap is a collection that stores key-value pairs and allows fast lookups using hashing.

29. **What is the difference between HashMap and TreeMap?**

    Answer: HashMap does not maintain order, while TreeMap stores keys in sorted order.

30. **What is the Set interface in Java?**

    Answer: A collection that does not allow duplicate elements.

Multithreading

31. **What is multithreading in Java?**

    Answer: Multithreading is the concurrent execution of two or more threads.

32. **What is the difference between a process and a thread?**

    Answer: A process is an independent unit of execution, while a thread is a lightweight unit of execution within a process.

33. **How do you create a thread in Java?**

    Answer: By extending the Thread class or implementing the Runnable interface.

34. **What is synchronization in Java?**

    Answer: Synchronization is the process of controlling access to shared resources to prevent concurrency issues.

35. **What is the difference between wait() and sleep() in Java?**

    Answer: wait() is used for inter-thread communication and releases the lock, while sleep() pauses the thread for a specified time without releasing the lock.

Java 8 Features

36. **What is a lambda expression in Java?**

    Answer: A lambda expression is a concise way to represent an anonymous function (i.e., a function without a name).

37. **What are functional interfaces in Java?**

    Answer: An interface with a single abstract method, used to define lambda expressions.

38. **What is the Stream API in Java?**

    Answer: The Stream API is used for processing sequences of elements (e.g., collections) in a functional style.

39. **What is the default keyword in Java 8?**

    Answer: It allows the definition of default methods in interfaces.

40. **What is the Optional class in Java 8?**

    Answer: Optional is a container object used to represent a value that may or may not be present.

Java Memory Management

41. **What is the heap memory in Java?**

    Answer: The heap is the memory area where objects are dynamically allocated.

42. **What is the stack memory in Java?**

    Answer: The stack is used for storing method calls, local variables, and references to objects in the heap.

43. **What is garbage collection in Java?**

    Answer: Garbage collection is the automatic process of reclaiming memory by deleting unreachable objects.

44. **What are the different types of garbage collectors in Java?**

    Answer: Serial GC, Parallel GC, CMS (Concurrent Mark-Sweep), and G1 GC (Garbage-First).

45. **What is the finalize() method in Java?**

    Answer: finalize() is called by the garbage collector before an object is destroyed.

Java Best Practices

46. **What is method overloading in Java?**

    Answer: Method overloading is defining multiple methods with the same name but different parameter types.

47. **What is method overriding in Java?**

    Answer: Method overriding is redefining a method in a subclass that already exists in the superclass.

48. **What is the use of the super keyword in Java?**

    Answer: super refers to the superclass of the current object.

49. **What is the use of the this keyword in Java?**

    Answer: this refers to the current instance of the class.

50. **What is a singleton class in Java?**

    Answer: A singleton class ensures that only one instance of the class is created.

Advanced Java Concepts

51. **What is the difference between == and equals() when comparing objects in Java?**

    Answer: == compares object references, while equals() compares the actual content of the objects.

52. **What is the transient keyword in Java?**

    Answer: It marks a member variable as not serializable during serialization.

53. **What is the volatile keyword in Java?**

    Answer: It ensures that a variable's value is always read from and written to the main memory, preventing caching by threads.

54. **What is the difference between StringBuilder and StringBuffer?**

    Answer: Both are used for mutable strings, but StringBuilder is faster as it is not synchronized, while StringBuffer is synchronized and thread-safe.

55. **What is the super keyword used for in Java?**

    Answer: It refers to the superclass and is used to access superclass methods and constructors.

56. **What are synchronized methods in Java?**

    Answer: Methods that are synchronized ensure that only one thread can access them at a time, to avoid concurrency issues.

57. **What is the difference between final, finally, and finalize?**

    Answer: final is used to define constants, finally is a block that runs after a try-catch block, and finalize is called before an object is garbage collected.

58. **What are Checked and Unchecked exceptions in Java?**

    Answer: Checked exceptions are checked at compile time (e.g., IOException), while unchecked exceptions are checked at runtime (e.g., NullPointerException).

59. **What is the purpose of the instanceof operator?**

    Answer: It checks whether an object is an instance of a specific class or subclass.

60. **What is the instance keyword in Java?**

    Answer: There is no instance keyword in Java; however, instanceof checks the instance type.

Java Memory and Performance

61. **What is the role of the Garbage Collector in Java?**

    Answer: The Garbage Collector automatically reclaims memory used by objects that are no longer reachable.

62. **What is the difference between String and StringBuffer in Java?**

    Answer: String is immutable, while StringBuffer is mutable and designed for efficient string manipulation.

63. **What is memory leak in Java?**

    Answer: A memory leak occurs when objects are no longer in use but are still referenced, preventing garbage collection.

64. **What are Stack and Heap in Java memory management?**

    Answer: The stack stores method frames, local variables, and references, while the heap stores objects and arrays.

65. **What is method overloading and method overriding?**

    Answer: Method overloading occurs when two or more methods have the same name but different parameters, while method overriding is when a subclass provides its own implementation of a method defined in the superclass.

66. **What is the role of volatile keyword in Java?**

    Answer: The volatile keyword ensures that the value of a variable is always updated across all threads.

67. **What is the difference between shallow copy and deep copy in Java?**

    Answer: A shallow copy copies references to objects, while a deep copy copies the objects themselves, creating a new instance.

68. **What is the difference between ArrayList and LinkedList in Java?**

    Answer: ArrayList provides fast random access but slow insertions and deletions, while LinkedList provides fast insertions and deletions but slower access.

69. **What is the difference between HashMap and TreeMap in Java?**

    Answer: HashMap does not maintain order of its keys, while TreeMap stores keys in a sorted order.

70. **What are the main methods in the Map interface?**

    Answer: put(), get(), remove(), containsKey(), containsValue(), and keySet().

Java Streams and Lambdas

71. **What is the difference between Stream and Collection in Java?**

    Answer: Stream is a sequence of elements supporting aggregate operations, while Collection is a framework for storing data.

72. **What is a Lambda expression in Java?**

    Answer: A lambda expression is a concise way to represent an anonymous function.

(a, b) -> a + b

73. **What is method reference in Java?**

Answer: Method reference is a shorthand notation for calling a method using the :: syntax.

ClassName::methodName

74. **What are the main operations in Java Streams?**

Answer: map(), filter(), reduce(), collect(), forEach(), and sorted().

75. **What is the purpose of Collectors in Java?**

Answer: Collectors is a utility class used with the Stream API to collect results (e.g., toList(), joining(), groupingBy()).

76. **What is Optional in Java?**

Answer: Optional is a container object which may or may not contain a non-null value, helping to avoid NullPointerException.

77. **What is the use of the Function interface in Java?**

Answer: The Function interface is used to represent functions that take an argument and return a result.

78. **What is the use of the Predicate interface in Java?**

Answer: The Predicate interface represents a condition that returns a boolean value.

79. **What is the difference between map() and flatMap() in Java?**

Answer: map() transforms elements into a single value, while flatMap() transforms each element into a sequence of values, flattening them.

80. **What is reduce() in Java Streams?**

Answer: reduce() is a terminal operation that combines elements of a stream into a single result.

Concurrency and Threading

81. **What is the difference between a Thread and a Runnable in Java?**

Answer: A Thread is a class, and Runnable is an interface. You implement Runnable when you want to define a task for a thread to execute.

82. **What is the synchronized keyword in Java?**

Answer: synchronized ensures that only one thread can access a method or block of code at a time.

83. **What are Executor and ExecutorService in Java?**

Answer: Executor is an interface for managing thread execution, and ExecutorService is a subclass that provides methods for managing tasks.

84. **What is the difference between wait() and sleep() in Java?**

Answer: wait() releases the lock and pauses execution, while sleep() pauses execution without releasing the lock.

85. **What are CountDownLatch and CyclicBarrier in Java?**

Answer: CountDownLatch blocks a thread until a specified number of events occur, and CyclicBarrier synchronizes threads at a barrier point.

86. **What is a deadlock in Java?**

Answer: A deadlock occurs when two or more threads are blocked indefinitely, waiting for each other to release resources.

87. **What is the difference between notify() and notifyAll() in Java?**

Answer: notify() wakes up one thread waiting on an object, while notifyAll() wakes up all threads waiting on that object.

88. **What is ThreadLocal in Java?**

Answer: ThreadLocal provides thread-local variables, where each thread has its own copy of the variable.

89. **What is the purpose of volatile keyword in Java?**

Answer: It ensures that changes to a variable are always visible to other threads.

90. **What is ForkJoinPool in Java?**

Answer: ForkJoinPool is a specialized thread pool for parallelizing tasks that can be broken down into smaller sub-tasks.

Advanced Java Concepts

91. **What is reflection in Java?**

Answer: Reflection is a feature that allows runtime inspection and modification of class properties, methods, and fields.

92. **What is the Proxy class in Java?**

Answer: Proxy allows creation of dynamic proxy classes that implement interfaces at runtime.

93. **What is a Serializable interface in Java?**

Answer: Serializable is a marker interface used to indicate that an object can be converted into a byte stream for storage or transmission.

94. **What is enum in Java?**

Answer: An enum is a special class representing a group of constants (e.g., days of the week).

95. **What is the volatile keyword used for in Java?**

Answer: It is used to ensure that updates to a variable are visible to all threads immediately.

96. **What is dependency injection in Java?**

Answer: Dependency injection is a design pattern where an object's dependencies are provided to it rather than created by the object itself.

97. **What are Java annotations?**

Answer: Annotations provide metadata about the code and are used for configuration or compile-time checks.

98. **What is the difference between == and .equals() in Java?**

Answer: == compares references, and .equals() compares the actual contents of objects.

99. **What is the purpose of finalize() method in Java?**

Answer: The finalize() method is called by the garbage collector before an object is destroyed.

100. **What is a class loader in Java?**

Answer: A class loader is responsible for loading classes into memory during runtime.