Java Top Interview Questions :

Theoretical Questions

1. What is Java, and what are its main features?
2. Explain the concept of Object-Oriented Programming (OOP) in Java.
3. What is the difference between JDK, JRE, and JVM?
4. What are the access modifiers in Java, and how do they work?
5. What is the difference between `==` and `equals()` in Java?
6. Explain the concept of inheritance and its types.
7. What is polymorphism, and how is it implemented in Java?
8. Describe encapsulation and its benefits.
9. What are the different types of exceptions in Java?
10. What is the purpose of the `final` keyword in Java?
11. What is a Java interface, and how is it different from an abstract class?
12. Explain method overloading and method overriding.
13. What is the Java Collections Framework?
14. Describe the differences between `ArrayList` and `LinkedList`.
15. What are generics in Java?
16. Explain the concept of multithreading and synchronization in Java.
17. What is garbage collection in Java, and how does it work?
18. What are the differences between `String`, `StringBuilder`, and `StringBuffer`?
19. Explain the concept of Java annotations.
20. What is the purpose of the `transient` keyword?

Coding Questions

21. Write a program to reverse a string.
22. Write a Java function to check if a number is prime.
23. Implement a Java program to find the factorial of a number using recursion.
24. Write a program to count the frequency of each character in a string.
25. Write a function to find the maximum and minimum elements in an array.
26. Write a Java program to check if a string is a palindrome.

27. Implement a method to find the second largest number in an array.
28. Write a program to remove duplicates from an array.
29. Implement a binary search algorithm in Java.
30. Write a Java program to sort an array using bubble sort.
31. Create a program to merge two sorted arrays.
32. Write a Java program to find the Fibonacci series up to `n` numbers.
33. Implement a program to find the common elements in two arrays.
34. Write a method to check if two strings are anagrams.
35. Write a Java program to find the longest substring without repeating characters.
36. Implement a function to find the intersection of two linked lists.
37. Write a program to rotate an array by `k` positions.
38. Implement a method to find the middle node of a linked list.
39. Write a program to detect a cycle in a linked list.
40. Merge two sorted linked lists into one sorted linked list.

Theoretical Questions (Continued)

41. What is the difference between a shallow copy and a deep copy?
42. Explain the concept of immutability in Java.
43. What are lambda expressions, and how are they used in Java?
44. What is the purpose of the `default` keyword in interfaces?
45. What is the `static` keyword, and how is it used in Java?
46. Explain the concept of method references in Java.
47. What is the purpose of the `volatile` keyword in Java?
48. What are the different types of inner classes in Java?
49. What is a constructor in Java, and how does it differ from a method?
50. Describe the Java Memory Model and its importance.
51. What is the difference between `throw` and `throws` in Java?
52. What is a `Map` in Java, and how does it differ from a `Set`?
53. Explain the concept of stream processing in Java 8.
54. What is the purpose of the `synchronized` keyword?
55. What is the difference between a stack and a queue?
56. Explain the concept of dependency injection in Java.

57. What is the significance of the `this` keyword?
58. Describe how to implement a Singleton pattern in Java.
59. What are the advantages of using Java frameworks like Spring?
60. Explain the concept of microservices architecture.

Coding Questions (Continued)

41. Write a program to implement a simple calculator (addition, subtraction, multiplication, division).
42. Write a Java program to check if a given string is a valid parenthesis sequence.
43. Create a program to find the missing number in an integer array containing `n` distinct numbers.
44. Implement a method to find the longest common prefix among an array of strings.
45. Write a program to find the sum of all even numbers in an array.
46. Create a program to find all permutations of a string.
47. Implement a binary tree and write methods for in-order, pre-order, and post-order traversal.
48. Write a program to find the height of a binary tree.
49. Create a function to calculate the nth Fibonacci number using dynamic programming.
50. Implement a program to check for balanced brackets in an expression.
51. Write a program to reverse a linked list in groups of `k`.
52. Create a program to find the union of two arrays.
53. Implement a method to find the longest increasing subsequence in an array.
54. Write a program to rotate a linked list.
55. Create a function to find the diameter of a binary tree.
56. Write a program to implement a basic priority queue.
57. Implement a method to detect a cycle in a directed graph.
58. Write a program to perform level-order traversal of a binary tree.
59. Create a program to flatten a nested list of integers.
60. Implement a function to convert a binary search tree to a sorted doubly linked list.

Theoretical Questions (Continued)

61. What is the difference between `==` and `equals()` in Java?
62. Explain the concept of autoboxing and unboxing in Java.
63. What are design patterns, and why are they useful?
64. What is the difference between a `HashMap` and a `TreeMap`?

65. Explain the concept of garbage collection in Java.
66. What is a `try-with-resources` statement?
67. Describe how to handle exceptions in Java.
68. What are functional interfaces in Java?
69. Explain the significance of the `final`, `finally`, and `finalize` keywords.
70. What are generics in Java, and how do they improve type safety?
71. What is method overriding, and how does it differ from method overloading?
72. Describe how Java handles multithreading.
73. What is the purpose of the `instanceof` operator?
74. Explain the difference between a `constructor` and a `method`.
75. What is the Java Collections Framework?
76. How can you create an immutable class in Java?
77. Explain the concept of a `HashSet` and how it differs from a `List`.
78. What is the role of the Java Runtime Environment (JRE)?
79. What is the difference between the `StringBuilder` and `StringBuffer` classes?
80. What are the access modifiers in Java, and how do they differ?

Coding Questions (Continued)

61. Write a program to implement a binary search tree.
62. Create a function to find the maximum product of two integers in an array.
63. Write a program to check if a linked list is a palindrome.
64. Implement a method to find the longest substring without repeating characters.
65. Write a program to find the intersection of two arrays.
66. Create a function to determine if a number is prime.
67. Implement a method to perform a selection sort on an array.
68. Write a program to remove duplicates from a sorted linked list.
69. Create a program to implement a queue using an array.
70. Write a program to find the first repeating character in a string.
71. Implement a method to calculate the factorial of a number using recursion.
72. Write a program to find the largest prime factor of a number.
73. Create a function to check if two strings are rotations of each other.

74. Implement a method to find the common elements in three sorted arrays.
75. Write a program to convert a Roman numeral to an integer.
76. Create a function to check if a string is a valid palindrome (ignoring spaces and punctuation).
77. Implement a method to reverse the words in a given string.
78. Write a program to check if an array contains duplicates within a given range.
79. Create a function to find the shortest path in an unweighted graph using BFS.
80. Implement a method to merge two sorted arrays into one sorted array.

Theoretical Questions (Continued)

81. What are the different types of exceptions in Java?
82. Explain the significance of the `default` keyword in Java interfaces.
83. What is the Observer design pattern, and where is it used?
84. Describe the difference between `ArrayList` and `LinkedList`.
85. What is the purpose of the `java.util.Optional` class?
86. Explain the concept of method chaining in Java.
87. What is the use of the `volatile` keyword in multithreading?
88. What is the difference between shallow and deep cloning in Java?
89. Describe the purpose of the `transient` keyword in serialization.
90. What is the difference between a stack overflow and a stack underflow?

Coding Questions (Continued)

81. Write a program to find the second largest number in an array.
82. Implement a method to calculate the power of a number using recursion.
83. Write a program to find all anagrams of a given string in a list.
84. Create a function to find the longest common substring between two strings.
85. Implement a program to generate the first `n` Fibonacci numbers.
86. Write a program to check if a given binary tree is a balanced tree.
87. Create a method to rotate a matrix 90 degrees clockwise.
88. Implement a function to find all unique triplets in an array that sum to zero.
89. Write a program to convert a binary number to decimal.
90. Create a function to find the median of two sorted arrays.

Theoretical Questions (Continued)

91. What are the key principles of object-oriented programming?
92. Describe the process of class loading in Java.
93. What is the purpose of the `assert` keyword in Java?
94. What is the Java Development Kit (JDK)?
95. Explain the difference between `throw` and `throwable` in Java.
96. What is a `Future` in Java concurrency?
97. What are the main advantages of using Java?
98. Explain the concept of polymorphism with examples.
99. What is the purpose of the `static` block in Java?
100. Describe the significance of the `new` keyword in Java.

Coding Questions (Continued)

91. Write a program to find the longest repeating character sequence in a string.
92. Implement a method to flatten a multi-dimensional array.
93. Create a program to find the smallest missing positive integer in an array.
94. Write a function to find the longest palindromic substring in a string.
95. Implement a program to perform a binary search on a rotated sorted array.
96. Write a program to find the count of prime numbers less than a given number.
97. Create a function to compute the GCD (Greatest Common Divisor) of two numbers.
98. Implement a method to check if a number is a power of three.
99. Write a program to generate all subsets of a set.
100. Create a function to check if a binary tree is symmetric.

Theoretical Questions (Continued)

101. What is the concept of "fail-fast" in Java collections?
102. Explain the difference between `==` operator and `.equals()` method in the context of strings.
103. What is the purpose of the `super` keyword in Java?
104. What is an interface in Java, and how is it different from an abstract class?
105. Describe the concept of `garbage collection` and its algorithms.

106. What are the access levels provided by access modifiers in Java?

107. Explain the concept of a `callback` in Java.

108. What is the difference between `throwable` and `exception`?

109. What is `Java Native Interface (JNI)`?

110. Describe the role of a Java `ClassLoader`.

Coding Questions (Continued)

101. Write a program to find the common elements in two unsorted arrays.

102. Implement a method to calculate the sum of all odd numbers in an array.

103. Write a program to implement a binary search on a sorted linked list.

104. Create a function to generate all combinations of parentheses.

105. Implement a program to check if a number is a happy number.

106. Write a method to check for duplicate characters in a string without using extra space.

107. Create a program to find the longest consecutive sequence in an array.

108. Write a function to calculate the square root of a number using binary search.

109. Implement a program to find the first non-repeating character in a string.

110. Create a method to count the number of vowels and consonants in a string.

Theoretical Questions (Continued)

111. What are the differences between JDK, JRE, and JVM?

112. Describe the concept of multithreading and its advantages.

113. What is a thread pool in Java?

114. What is the purpose of the `join()` method in Java threads?

115. Explain the term `immutable` with respect to strings in Java.

116. What is an `enum` in Java, and how is it used?

117. Describe the difference between `private`, `protected`, and `public` access modifiers.

118. What are the uses of the `synchronized` keyword?

119. What is the purpose of the `native` keyword in Java?

120. Explain the significance of `Object` class in Java.

Coding Questions (Final Set)

111. Write a program to implement the bubble sort algorithm.

112. Create a method to find the missing number in an arithmetic progression.

113. Write a function to determine if a string contains all unique characters.

114. Implement a method to find the maximum sum of a contiguous subarray.

115. Create a program to check if two strings are anagrams of each other.

116. Write a program to find the number of words in a given string.

117. Implement a method to find the first non-repeating character in a string using a hash map.

118. Create a program to determine the length of the longest substring without repeating characters.

119. Write a function to find the lowest common ancestor of two nodes in a binary search tree.

120. Implement a program to perform a depth-first search on a graph.