1. Core Java

- - What are the features of Java?
- - What is OOPs? Explain with real-life examples.
- - What is the JVM, JRE, and JDK?
- - Explain the Java memory model.
- - What is the difference between == and .equals()?
- - What are primitive and reference data types?
- - Call by Value and Call by Reference?
- - What is type casting in Java?
- - What are access modifiers in Java?
- - What is the static keyword used for?
- - Can static method override?
- - What is the final keyword?
- - What is the difference between final, finally, and finalize()?
- - What is method overloading and overriding?
- - What is the difference between compile-time and runtime polymorphism?
- - What is encapsulation in Java?
- - What is abstraction?
- - What is inheritance? Explain single, multilevel, and hierarchical inheritance.
- - Why is multiple inheritance not supported with classes?
- - Can Java support multiple inheritance? How?
- - What is a constructor? Types of constructors?
- - What is constructor chaining?
- - What is the use of this and super keyword?
- - What is an enum?
- - What is a marker interface?
- - What is the transient keyword?
- - What is the volatile keyword?
- - Difference between String, StringBuilder, and StringBuffer?
- - How is memory managed in Java?
- - What is garbage collection?
- - What is a package?
- - What is the difference between this and super keyword?
- - What is the use of the synchronized keyword in static methods?
- - Difference between shallow copy and deep copy?
- - What is cloning in Java?
- - What are the four pillars of OOP?
- - Explain IS-A and HAS-A relationships?

2. Collections and Generics

- - What is the Collection framework in Java?
- - What is the difference between List, Set, and Map?
- - Difference between ArrayList and LinkedList?
- - Difference between HashSet, TreeSet, and LinkedHashSet?
- - How does HashMap work internally?
- - What is the load factor and initial capacity?
- - Difference between HashMap and Hashtable?
- - Difference between HashMap and ConcurrentHashMap?
- - What are fail-fast and fail-safe iterators?
- - Difference between Comparable and Comparator?
- - How to sort a Map by keys or values?
- - What are generics in Java?
- - What is type erasure?

3. Multithreading and Concurrency

- - What is a thread in Java?
- - How to create threads?
- - Difference between Runnable and Thread?
- - What is the lifecycle of a thread?
- -What is the difference between `sleep()` and `wait()`?
- -What is the difference between `start()` and `run()` method?
- -What is join() method in thread?
- -What is the use of `yield()`?
- - What is synchronization?
- - What is volatile keyword?
- - What is ThreadLocal?
- - What is the use of wait(), notify(), and notifyAll()?
- - What is a daemon thread?
- - What is a deadlock? How to prevent it?
- - What is thread priority?
- - What is ExecutorService?
- -Why use ExecutorService instead of raw Threads?
- -What are the different types of Thread Pools in ExecutorService?
- -Difference between submit() and execute()?
- -What is a Future in ExecutorService?
- -How to handle exceptions in ExecutorService?
- -When to use CachedThreadPool vs FixedThreadPool?
- -How to wait for all tasks to complete?
- -Can ExecutorService execute tasks in order?
- - Difference between Callable and Runnable?

- - What is the difference between synchronized block and method?

4. Exception Handling

- - What is the difference between checked and unchecked exceptions?
- - How do you create a custom exception?
- - What is the use of finally block?
- - Can we write try without catch or finally?
- - What is try-with-resources?
- - What is the difference between throw and throws?
- - What is multi-catch block?

5. Java 8 and Functional Programming

- - What is a lambda expression?
- - What are functional interfaces?
- - What are default and static methods in interface?
- - What is Stream API?
- - Difference between map and flatMap?
- - What is Optional and how to use it?
- - What are method references?
- - Explain method reference and constructor reference?
- - New Date and Time API (LocalDate, LocalTime, etc.)
- - What is the difference between Class.forName() and new keyword?
- - What are annotations and how to create?
- -What is reflection and how to use?

6. Java Memory Management

- - What is JVM Architecture?
- - Heap vs Stack memory?
- - What are classloaders in Java?
- - What are different memory areas in JVM?
- - What are different GC algorithms?
- - What is memory leak?
- - How garbage collection works in Java?
- - What are strong, weak, soft, and phantom references?

7. Java I/O and Serialization

- - What is serialization?
- - What is deserialization?

- - What is transient keyword?
- - What is the difference between FileReader and BufferedReader?
- - What is the difference between byte stream and character stream?
- - What is ObjectOutputStream and ObjectInputStream?

8.Advance Question in Java

1. Why String is immutable in Java ?
2. Difference between String and StringBuffer in java ?
3. Can we override Static method in Java ?
4. Is string thread safe ?
5. What is String constant pool in java ?
6. How to prevent your class from being subclassed ?
7. Which two methods are overridden by an Object, intended to be used as key in Hashmap ?
8. Difference between List and Set in Java ?
9. How do you make a class immutable in java ?
10. Difference between Hashtable and Hashmap in java ?
11. How will you iterate map ?
12. What is type safety ? How to use it ?
13. Difference between ArrayList and LinkedList in java ?
14. Difference between checked and unchecked exception in java ?
15. How to convert array to a list ?
16. Difference between Serializable and Externalizable interface in java ?
17. Explain Singleton Design Pattern ? write you class to make singleton design pattern
18. Difference between local and instance variable
19. What is serialization

20. Difference between == and .equals in java ?

21. Difference between == and .equals in java ?

22. Difference between unique key and primary key ?

23. Difference between method overriding and method overloading in java ?

24. What are types of access specifiers

25. Explain use of this keyword ?

26. Explain exception hierarchy ?

27. What is index and view in database ?

28. Difference between extending thread class and runnable interface ?

29. How can we avoid null pointer exception ?

30. OOPS concepts

31. Difference between Abstraction and encapsulation ? how you use it in your project

32. When to use abstract class and interface in java ?

33. Difference between abstract class and interface ?

34. Difference between Exception and errors ?

35. Difference between throw, throws and finally

36. What is class ?

37. What is an object ?

38. Difference between class and object ?

39. What is method overloading

40. What is method overriding

41. What is a thread

42. File handling in Java ?

43. Purpose of static method and static variable ?

44. Difference between continue and break

45. What is final keyword in java ?

46. What is JIT compiler ?

47. What is WORK concept in java ?

48. Purpose of default constructor ?

49. What is class loader in java ?

50. What is inheritance

51. What is object cloning

52. What is method overriding

53. Static binding and dynamic binding

54. What is interface

55. What is exception handling in java ?

56. Can finally block be used without catch in java ?

57. What is garbage collection?

58. Difference between Array and vector ?

59. Why don't we have destructors in Java ?

60. What is object and how it is stored in Java ?

61. Explain JDK, JRE and JVM in java ?

62. Why char array is preferred to store password than Strings ?

63. How many objects are created in JVM when a string type variable is assigned using new keyword ?

64. What is classpath in java ?

65. Is java completely object oriented ?

66. What makes java platform independent ?

67. What is dynamic polymorphism ?

68. What is multiple inheritance in java ?

69. Is multiple inheritance supported in java ?

70. What is a wrapper class in java ?

71. Can a java file have more than one public class ?

72. Write a program to remove duplicates from an array ?

73. Program to reverse a string without using any api ?

74. What design patterns you have used in your app ?

75. What are the tools you have used to improve the performance of your application ?

76. How you optimize the code for jdbc

77. What are the differences between C++ and Java

78. List the features of the Java Programming language

79. What is a ClassLoader

80. What are the Memory Allocations available in JavaJava

81. What are the differences between Heap and Stack Memory in Java?

82. Define Copy Constructor in Java

83. Can you implement pointers in a Java Program

84. Differentiate between instance and local variables

85. Can you call a constructor of a class inside another constructor

86. How does the size of ArrayList grow dynamically?

87. Difference between static methods, static variables, and static classes in Java

88. Why is Java, not a pure object-oriented language

89. What do you understand about an instance variable and a local variable?

90. What are Composition and Aggregation

91. What do you mean by data encapsulation

92. What is the 'IS-A' relationship in OOPs Java

93. How is the creation of a String using new() different from that of a literal

94. Why is synchronization necessary

95. What are shallow copy and deep copy in Java

96. Write a Java Program to print Fibonacci Series using Recursion

97. Write a Java Program to find the factorial of a given number

98. Does Java work as a "pass by value" or "pass by reference" phenomenon

99. What is the difference between systems.out, System.err, and System.in?

100. What is the purpose of generics ?

101. In what situations is the "super" keyword used ?

102. How to write multiple catch statements under a single try block

103. What is casting ?

104. What is implicit casting ?

105. What is explicit casting ?

106. What is an enum ?

107. Can you give example of different utility methods in string class ?

108. In which scenario, code in finally is not executed ?

109. How do you create custom exception ?

110. Is try without catch is allowed ?

9.Practical in Java

1.reverse string

2.check for palindrome

3.fibonacci series

4.factorial number check

5.count vowels and consonants

6.sort an array

7.merge two array

8.find largest element in an array

9.remove duplicate from an array

10.check if a number is Armstrong

11.reverse a number

12.calculate GCD of two number

13.prime number check

14.check for anagram

15.count the number of digits in a number

16.print the prime numbers in a range

17.find the second largest element in an array

18.swap two number

19.print the pascals triangle

20.find the missing number in an array

21.convert decimal to binary

22.check for perfect number

23.find the sum of digits of a number

24.find the length of a string

25.check if a string is empty

26.count the occurrences of a character in a string

27.find the first non-repeated character in a string

28.remove all white space from a string

29.find the common elements in two array.

30.find the factorial of a number using recursion

31.generate random number

32.check if a year is leap year

33.find the sum of first N natural numbers

34.check if a string contains another string

35.find the maximum occurring character in a string

36.implementing bubble sort

37.implementing selection sort

38.check whether a list of integers contains only ODD numbers.

39.check whether the two integer arrays contain the same elements.

40.rotate an array in left direction by ‘n’ positions.

41. reverse words in String like lastWord = "Hello welcome" should be “olleH emoclew”.

42. move all zeros in an array to the end

43. Perform Two sum approach

44. Find common elements in array

45. Merge two array and sort using stream.

10.Thread Question in Java

1.Write a program to create two threads: one prints even numbers, the other prints odd numbers.

2.Implement a producer-consumer problem using wait() and notify().

3.Write a program to create a deadlock situation.

4.Demonstrate thread-safe singleton class.

- **Without Singleton:** Each thread creates its own instance → wasteful and unsafe for shared resources.

- **With Thread-Safe Singleton:** Only one instance across threads → efficient and consistent.

5.Use Callable and Future to return a result from a thread.

6.Write a program to implement thread pool using ExecutorService.

7.Demonstrate usage of volatile variable.

8.Write a program where multiple threads update a shared counter safely.

9.Print numbers from 1 to 100 using multiple threads in sequence.

10.Demonstrate daemon thread behavior with example.