1. What is an abstract class? How is it different from an interface with default methods?
2. What are the rules for Method Overloading and Method Overriding?
3. What is a marker interface? Name some marker interface and explain their purpose/utility.
4. What is a Functional Interface? What kind of components can a functional interface contain?
5. What is an inner interface? Can you name any inner interfaces in the standard JDK?
6. What is an inner class? How do you declare a variable & create object of an inner class?
7. What is an anonymous inner class?
8. What is a Lambda expression?
9. How is an anonymous inner class different from a Lambda expression?
10. Why doesn’t Java support multiple inheritance? [What is diamond problem in Java? How is this problem resolved in Java?]

<https://www.javatpoint.com/what-is-diamond-problem-in-java>

1. What are the different ways to declare constants in Java?
2. Explain the different usages of the keyword *final*.
3. Differences between *final, finally* and *finalize*.
4. Explain the difference uses of final keyword.
5. Can constructors, static methods and private methods be overridden? Why?
6. Explain the exception handling mechanism in Java.
7. Difference between error and exception. What is Throwable?
8. Explain the concept of checked and unchecked exceptions.
9. How to create custom checked and unchecked exceptions?
10. Can we write a try block without catch block?
11. What are the different options available to handle multiple exceptions for a given try-block?
12. How to handle multiple exceptions together? [How to handle multiple exceptions the same way?]
13. Can we throw an exception in a finally-block?
14. Difference between throw and throws keyword. And discuss Exception Propagation
15. Explain rules related to Exception handling w.r.t. method overriding.
16. Explain try-with-resources
17. When to use String/StringBuffer/StringBuilder?
18. Explain equals and hashCode contract
19. What is Comparable and Comparator?
20. How to compare a list of Employees based on designation and age such that if designation of the employee is same then sorting should be based on age?
21. Explain different methods of Object class
22. Explain System.out.println() statement
23. **System** is a class in java.lang package
24. **out** is a static attribute in the class System.
25. The attribute out is of type java.io.PrintStream which is a sub-class of java.io.OutputStream.
26. The PrintStream class has many overloaded versions of print and println methods.
27. Explain how variable arguments work. How many variable arguments can we have in a method signature and in which position?
28. What is a wrapper class? What is boxing and unboxing?
29. Explain Auto-boxing and auto-unboxing
30. Explain static keyword usage
31. When does a init block (static block) get executed? What is the use of init block (static block)?
32. What are the different types of constructors?
33. Explain constructor chaining.
34. What is Cloneable? Explain shallow cloning and deep cloning.
35. What is Serialization and De-serialization?
36. What is SerialVersionUID?
37. How to prevent data from being shared during serialization?
38. Difference between Serializable and Externalizable
39. How to make a class Immutable?
40. Explain Class loaders in Java

https://www.geeksforgeeks.org/classloader-in-java/

1. **What is Singleton Design Pattern and how can it be implemented?**
2. **What is a Factory Pattern and Abstract Factory Pattern?**
3. **What are the different types of Design Pattern and why do we use them?**
4. What is Garbage Collection in Java? When is an object eligible for garbage collection?
5. Explain Generics in Java
6. What is Enum?
7. Can Enum extend another class
8. Can Enum implement interfaces?
9. Explain how Arrays.binarySearch() works.
10. Explain the difference between linear search and binary search.
11. How to convert an array into a stream?
12. How is a stream different from a collection?
13. Explain different intermediary and terminal operations in Stream API.
14. Explain collect() method of Stream API? Is collect() an intermediary or a terminal operation?

**Coding exercises:**

1. Calculation of factorial.
2. Calculation of sum of factorials of 1 to N.
3. Calculation of exponent (use BigInteger)
4. Swap two numbers – with / without using a temporary variable
5. Find max / min of 3 numbers without swapping
6. Find max / min of n numbers with swapping
7. Code to check whether a number is prime number or not
8. Reverse the characters in a string without using reverse method
9. Declare and initialize a 2D array of 3-by-3 size and display the array contents using nested for-loop.
10. Write the code for a method that accepts a 2D array of int values and returns the average of all the values in that 2D array.
11. Problems checking knowledge of BODMAS rule (precedence of arithmetic operators).
12. Problems checking knowledge of precedence of logical operators.