

1. What is a package in Java? What's the actual need for it?
2. What is Class?
3. What is an Object? How is it different from Class?
4. What is an access modifier in Java? What are they & what's the use of it?
5. Overloading
6. What are primitive types?
7. String immutable, SCP & String intern();
8. Read Java API docs - [Object](#) (1.8 or jdk8). What's the significance of this class?
9. Object hashCode & equals methods importance & details.
10. Importance of toString() method of Object class.
11. try-catch-finally
12. Closeable & Autocloseable
13. try with resources.
14. Exception Handling in detail.
 1. Throwable
 2. java.lang.Error
 3. java.lang.Exception
 4. Error vs Exception
 5. Checked & unchecked Exceptions
 6. Throws
 7. Chained Exceptions
 8. Cause
 9. User defined Exception or Custom Exception.
15. What is NullPointerException? Is it good to catch that exception?
16. What are instance variables & instance methods?
17. What are static variables & static methods? What's the significance?
18. [Inheritance](#)
19. [Interface & Default Methods](#).
20. Overriding. Explore about @override annotations as well.
21. Runtime and compile time polymorphism
22. Abstract class vs Class
23. Abstract class vs Interface.
24. Data Structures:

Hint: Refer [Java Docs](#) & usage:

 - a. [Collection](#)
 - b. [Arrays](#), [ArrayList](#), [Vector](#)
 - c. [Set](#), [Hashtable](#), [Properties](#), [HashMap](#)

d. [LinkedHashMap](#)

25. [Generics](#)

26. [Enum Types](#)

27. POJO class & it's significance.

28. Types of For Loops & syntax.

29. Object Cloning

30. BufferedReader vs InputStreamReader, BufferedWriter vs OutputStreamWriter

31. Inner Classes & Anonymous Classes

32. [Method Reference](#)

33. Autoboxing & Unboxing

34. Varargs

35. Reflection

36. [Regular Expression](#) in java ([Pattern](#) , Matcher etc)

37. Hashing Algorithms

38. Hashing internals of Hashmap.

39. Threads & Runnable

40. Synchronization & locks