



## Java

1. What is Encapsulation?
2. What is Abstraction?
3. What is Inheritance?
4. What is Polymorphism?
5. What is Casting?
6. How many objects on the heap if I cast?
7. What is an Interface?
8. Can I instantiate an Abstract class? Constructor?
9. Why would you use an Abstract class over an Interface?
10. equals() vs ==?
11. Can I force garbage collection?
12. Which method does the garbage collector call?
13. What is the finally block?
14. Is a catch block needed?
15. What are Generics for?
16. Comparable vs Comparator
17. Hashtable vs HashMap
18. How do I start a thread implementing Runnable?
19. Thread methods. (run, start, sleep, wait, notify, notifyAll, join).
20. Difference between the run and start method?
21. What is a deadlock?
22. Which is the fastest File IO available and why?
23. Which are the scopes of a variable?
24. Can you override static methods?
25. What is shadowing?
26. What are Wrapper classes?
27. What is Varargs used for?
28. What is the difference between protected and default?
29. What is the final keyword used for?
30. What is the difference between StringBuilder and Buffer?
31. What is synchronization?
32. How do you go about starting a thread?
33. What is the difference between a List of a Set?
34. Some concrete implementations of Set.
35. LinkedList vs ArrayList
36. How do you insert elements in a Map?

37. Different between Exception and Error?
38. What is Serialization?
39. What is a Singleton?
40. What is the IS-A rule?
41. Where are variable references stored?
42. When is an object ready for garbage collection?
43. What is reflection?
44. What makes the String class special?
45. What is the difference between an Exception and a RuntimeException?
46. Rules of the catch block?
47. What does the Iterable interface do?
48. HashSet vs TreeSet
49. What is a Map?
50. Can I sort a Map?
51. How do I start a Thread extending the Thread class
52. What is Starvation?
53. InputStream vs Reader
54. When is a class fully synchronized?
55. Can an Interface have variables? What are they?
56. What is an Abstract class?
57. What are default methods in an interface?
58. What's the first line in a constructor?
59. What is constructor chaining?
60. What is a short circuit operator?
61. Where are Strings stored in memory?
62. What is hashCode() for?
63. What's the parent of all exceptions?
64. Can I catch an error? Does it make sense?
65. Is there any case the finally block won't execute?
66. Array vs ArrayList
67. How do I insert elements in a map?
68. What is a Thread?
69. What is a Factory?
70. How do I make an object Serializable?

## SQL

1. All SQL Sub Languages.
2. Truncate vs Delete.
3. DATE vs TIMESTAMP.
4. What is Referential Integrity?
5. WHERE vs HAVING.
6. Scalar Functions.
7. INNER JOIN vs INTERSECT.

8. Can I do a sub query in an INSERT statement?
9. What does the 1NF say?
10. Properties of a transaction.
11. What is Atomicity?
12. What is a Cursor? What can I use it for?
13. Which are the main interfaces of JDBC?
14. ALTER vs UPDATE.
15. CLOB vs BLOB.
16. What is a Primary Key?
17. GROUP BY vs ORDER BY.
18. Aggregate Function.
19. FULL JOIN vs UNION.
20. Can I do a sub query in the FROM clause of a SELECT statement? Why?
21. What does the 2NF say?
22. What is Isolation?
23. What is the Read Committed Isolation level?
24. What is a Trigger? When can it execute?
25. What do I need to connect to a DB with JDBC?
26. CREATE vs INSERT.
27. What does a DATE subtraction return?
28. What is a Foreign Key?
29. WHERE vs HAVING.
30. Scalar Function.
31. LEFT JOIN vs MINUS.
32. Can I do a sub query in an UPDATE statement?
33. What does the 3NF say?
34. What is Consistency?
35. What is the Serializable Isolation level?
36. Stored Procedure vs UDF.
37. Statement vs PreparedStatement.
38. DROP vs DELETE.
39. VARCHAR vs VARCHAR2.
40. What is a composite Primary Key?
41. GROUP BY vs ORDER BY.
42. Scalar Function.
43. What is a SELF JOIN?
44. UNION vs UNION ALL.
45. Can I do a sub query in a DELETE statement?
46. What is Normalization?
47. What is Durability?
48. What is a Phantom read?
49. What is a Dirty read?
50. What is a Sequence?

51. Connection vs DriverManager.
52. INNER JOIN vs OUTER JOIN.
53. What is a theta join?
54. What is a natural join?
55. All types of joins. You can get asked differences between any of them.
56. What is an index?
57. Clustered vs Unclustered index.
58. What is a view?
59. Regular view vs Materialized view.
60. Unary, Binary and Ternary relations.
61. All sections of the SELECT statement.

## **Spring**

1. What is Spring?
2. Spring is loose coupled. Explain why do we say this.
3. Bean Scopes.
4. Spring Modules.
5. Bean Lifecycle.
6. What is the Spring Container?
7. ApplicationContext vs BeanFactory.
8. Inversion of Control.
9. Types of Dependency Injection
10. Different types of Autowiring.
11. Autowiring. How to configure it?
12. How do I create a bean in the container?
13. How do I enable/configure annotations?
14. Stereotype Annotations.
15. @Autowired.
16. @Autowired vs @Inject vs @Resource.
17. Contextual Session.
18. How do I integrate Spring and Hibernate?
19. How do I make Spring handle transactions for me? Configuration?
20. JSR-303.
21. JndiObjectFactoryBean.
22. What is Spring MVC?
23. Spring MVC Workflow.
24. Spring MVC Annotations.
25. What is @RequestMapping?
26. InternalResourceViewResolver.
27. ContextLoaderListener.
28. How to configure Spring MVC?
29. What is AOP?
30. How to configure AOP?

31. Cross Cutting Concerns.
32. Aspect.
33. Advice.
34. Advice Kinds.
35. @Before vs @Around.
36. @After vs @AfterReturning vs @AfterThrowing.
37. JoinPoint vs Point cut.
38. Spring AOP vs AspectJ.
39. Load time vs Compile time Weaving.
40. Path parameters.

## **Hibernate**

1. What is Hibernate?
2. What is ORM?
3. How do I set up an entity?
4. How do I relate entities? (Cardinality Mapping)
5. Lazy vs Eager.
6. How to create queries?
7. ACID properties.
8. Session vs SessionFactory.
9. L1 vs L2.
10. How do I configure Hibernate?
11. How do I map a class without annotations?
12. Difference between HQL and SQL.
13. Hibernate interfaces.
14. Session methods.
15. Difference between Criteria and HQL.
16. Object states.
17. What will Criteria or HQL return (list() of a POJO expected).
18. Default fetching strategy in Hibernate.
19. Hibernate exceptions.
20. How do you handle exceptions using Hibernate?
21. Get vs Load.
22. Update vs Merge.
23. All methods in the Session interface.
24. How do you configure ehcache?
25. Dynamic insert vs update.
26. Named queries.
27. How do you get an instance of the SessionFactory?
28. Can you perform joins in Hibernate?

## **JUnit**

1. What is your experience with JUnit?

2. Can you test your entire application with JUnit? (You would have to test all units of it [methods], but we normally separate this testing in different test cases (specific functionality) with multiple tests).
3. What does a Unit refer to?
4. Annotations.
5. Difference between @BeforeClass and @Before?
6. Why do we test software?
7. When does a test fail?
8. Different types of assert methods.
9. Which version of JUnit have you used?
10. How do you test your application automatically?