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 Serializible?

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 substraction 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 tetha 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.

Servlet

1. Explain the communication between the Client and the Server in a JEE application.
2. Servlet Hierarchy.
3. Life Cycle of Servlets.
4. Where do init, service and destroy come from in the hierarchy?
5. Init, Service and Destroy, how many times do they execute?
6. doGet vs doPost vs GET vs POST.
7. Another name for web.xml?
8. Sections and tags on the web.xml.
9. Servlet Context vs Servlet Config. How do you write it on the web.xml?
10. How do I declare a Servlet?
11. How does Error Handling work on JEE?
12. How does the Servlet Container track sessions?
13. Forward vs Redirect.
14. Where to forward from, Where to redirect from?
15. What is the service method signature on GenericServlet?
16. Service method in HttpServlet.
17. How many versions of init does HttpServlet have?
18. getAttribute vs getParameter.
19. Method signature doGet.
20. How to maintain user state?
21. What is the RequestDispatcher?
22. Front Controller.
23. What is a thread pool?
24. What are filters?
25. What is extended mapping?

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?
    a. Maven goals.

Web Services

1. What are Web Services?
2. Why do we use Web Services?
3. Protocol stack.
    a. Transport Protocol.
    b. Messaging Protocol.
    c. Description Protocol.
    d. Discovery Protocol.

4. How do you create a SOAP Web Service.
    a. Explain all the steps to expose and consume.
    b. Talk about contract first and contract last.
    c. Apache CXF for contract last.
5. How do you create a RESTful Web Service?
    a. Explain all the steps to expose and consume.
6. Annotations in REST.
7. How do you handle exceptions in REST?
8. What's the structure of the HTTP Request.
9. What's the structure of the HTTP Response.
10. What's the difference between @RestController and @Controller?
11. How do you handle exceptions in REST?
12. HTTP methods.
13. GET vs POST.
14. Idempotency.
15. What media types does REST support?
16. What does stateless mean?
17. Is SOAP stateless?
18. What are endpoints in SOAP?
19. WSDL Tags in detail.
20. What is the difference between portType and binding?
21. SOAP Binding styles.
22. SOAP message structure.
23. What is the header of the message used for?
24. What is the fault of the message used for? Where is it in the structure?
25. Contract First - Contract Last, how to implement it? (JAX-WS within Apache CXF)
26. SOAP exception handling.
27. Differences between SOAP and REST.
28. What media types does SOAP support?
29. What protocols does SOAP allow?

Angular

1. CLI commands to generate components and services.
    a) **ng generate component name**
    b) **ng generate service name**
2. When you start the angular application, what is the entry point of it, where does it start? (bootstrap).
3. What is a template?
4. Is it possible to have module.ts in every component you create?
    a) Not ideal, you should have one main module and then any custom module, like the routing module or anything that you want to modularize in your application.

5. How to inject a template into a view?
    a) (You normally do this with routing, and depending on a route the router will inject the template associated with a specific component).
6. Is it mandatory to export every component?
7. Explain the Http module.
8. What is Angular?
9. What is a Decorator?
10. What is a Module?
11. How do I create a Module, what does it receive?
12. How do I include a Module into my application?
13. What is a Component?
14. How do I create a component, what does it receive?
15. What is a Directive? Which are the types? Examples
16. What is Routing?
17. How do I inject an object into another object?
18. What is Data Binding?
19. Which are the types of Data Binding?
20. Syntax for all types of data binding.
21. What is a Pipe?
22. HTTP calls in Angular.
23. What is Observable?
24. Publisher/Subscriber.
25. Angular vs AngularJS.
26. How do I bundle my Angular application?
27. What is TypeScript?
28. What does TypeScript transpile into?
29. Access Modifiers in TypeScript.
30. Non Access Modifiers in TypeScript.
31. TypeScript data types.
32. Generics in TypeScript.

Microservices

1. Main advantages of using microservices.
2. Any disadvantage in using microservices.
3. Dependencies of Microservices (what do we need to use).
    a) Eureka (Discovery).
    b) Zuul (Gateway).
    c) Configuration Server.
    d) Kafka (Messaging).
    e) Hystrix (Monitoring).
    f) Docker (Orchestration).
4. Annotations of Eureka and what they do.
5. How to implement Registry and Discovery services.

6. What exception will get thrown if you try to run the client (single microservice) before the server (eureka)?
7. How to implement Gateway service.
8. What happens if you don't have a gateway? Can you have microservices without this?
9. Annotations of Zuul and what they do.
10. How do you make Zuul to employ discovery for the routes to services?
11. How to implement Monitoring service.
12. Annotations of Hystrix and what they do?
13. How to implement the Orchestration service?
14. What is Docker?
15. Workflow of a Docker? How does an image reach the container?
16. Explain each component in Docker containers.
17. Docker commands.
18. Image vs Instance.
19. Some properties in a docker-compose file.
20. How to make an image available in the main Docker instance.
21. Some commands in a docker file.
22. How do you create a docker image?
23. How to implement the Messaging Service.
24. Publisher/Subscriber in Kafka.
25. Annotations/Classes of Kafka and what they do?
26. How do we enhance the natural resiliency of the Microservices architecture?
27. Explain Spring Boot.
28. How do you configure the custom port in Spring Boot? (server.port = number in application.properties).
29. How do you reload any changes in Spring Boot without having to restart it?
30. How do you perform any database operations with Spring Boot?
31. What are actuators in Spring Boot?
32. What is Spring Data?
33. How do you define your custom Repository?

Soft Questions

- Tell me about yourself.
- Tell me about your most recent project.