* *Interview Questions Topic Wise*

***New Technologies:***

1. What is Kafka and how to configure it?
2. How to enable security for Kafka
3. What is Elastic Search? What is index, document, shards, and cluster?
4. How searching works internally in Elastic search?
5. RAML or Swagger
6. What is Blue Green Deployment
7. How does a load balancer work (explained with a diagram)
8. What is Devops, Jenkins puppet, chef?
9. Have you worked with any Performance Improving tools?
10. How to declare function in JavaScript?
11. How to call an AJAX call using JavaScript?
12. What is DOM
13. How to Synchronous and Asynchronous request in JavaScript.

***Basic Questions:***

1. Last project architecture.

***OOPS:***

1. OOPs features those u r working on
2. OOPS concepts – polymorphism, inheritance, encapsulation along with proper examples
3. SOLID Principles
4. ACID rule
5. What is Aggregation and composition?
6. Overriding with different scenarios (access specifier, return data type, exception).
7. Different between Encapsulation and Abstraction with examples.
8. Different between Overloading and Overriding with examples.
9. Different between Aggregation and Composition with examples.

***Design Patterns:***

1. Which design patterns have you implemented.
2. Design patterns u worked on & Adapter design pattern – how u implemented in project
3. If we have a Weather update service and 3 client want to have the data for weather for a city.
4. How can we implement this scenario using core java.(Intention to find which design pattern can be used)
5. How Factory pattern is achieved in spring.
6. Design Pattern (Adapter and Bridge)
7. Design a class diagram where Vehicle is parent and having two child class 2Wheeler and 4Wheeler

***OOAD:***

1. UML diagrams – class, sequence, component
2. Railway API design & many questions on it.

***Core Java:***

1. How would you create Immutable class and write down code
2. What is type inference(All version changes)
3. Write down singleton code and what is difference between java singleton and spring singleton
4. Hash map in detail.
5. Inheritance, composition, aggregation, examples & when to use
6. Collections , hashing technique, collection hierarchy, sorting and searching algorithms, comparable & comparator interfaces
7. Exception handling and hierarchy, difference between exception and error
8. Explain exception hierarchy?
9. Java SAX and DOM parser?
10. Concurrent HashMap vs HashMap
11. What is custom exception?
12. How we can sort a list of Employee class base on name and id. What is difference between Comparable and Comparator?
13. How many interfaces in Collections framework?
14. HashSet & Array List internal mechanism.
15. When to use Array List & when to use LinkedList
16. HashMap and Hashtable diff.?
17. What is Generics in java?
18. Hashcode and equals method
19. Array List & Liked list difference in depth
20. Linked list internal working
21. Reflections, generics and transactions
22. What is Hashing Techniques and do we need to override hash method for TreeSet.
23. Extends and super in Generics.

***Spring and Hibernate:***

1. How would you manage transaction in your project?
2. Spring security, spring mvc module?
3. Spring + Hibernate use of crud Repository.
4. What is LazyLoadingException? How can you avoid that?
5. What is the use of @EnableAutoConfiguration
6. Difference between @Component, @Service, @Repository annotation
7. How can I return json and xml from the same method of a controller
8. What is the use of @Transactional annotation why is it used
9. Should I specify @Transactional if the business is interacting with only one table
10. Is it feasible to specify @Transactional in controller, where should it be specified
11. What are Different level of Isolation in transaction management, which Isolation level you have used in your project?
12. Can we use request scoped bean into singleton bean? Have you worked on Persistence layer framework?
13. What is dispatcherServlet in Spring MVC?
14. How to achieve caching in spring.
15. Annotations and autowiring\*
16. Bean injections, inversion of control, Bean scope
17. What are the scope of the spring bean? And explain each.
18. Explain Spring IOC and DI.
19. Spring transactions
20. Spring MVC, springboot, Spring Validations
21. Hibernate or iBatis
22. Session factory and management of sessions
23. Advantage of ORM over JDBC connections
24. What are module you have used in spring?
25. Difference between @Component, @Service, @Repository and @Controller annotation.
26. What happened if you replace @Service and @Repository with @Component annotation?
27. How to handle transaction and at what layer we will use transaction.
28. What is Constructor-arg and Property in spring?
29. How you apply lazy loading in spring?
30. What is proxy in AOP? Have you used in your project?
31. How many types of hibernate cache?
32. Auto wiring in spring.
33. What are spring Interceptors (Write code for method implementation)
34. What are the spring Listeners and its types?
35. Bean Life Cycle.

***Rest Web Services and SOAP:***

1. Annotations used for creating REST service through Spring and Jersey
2. Difference between PUT, POST and PATCH?
3. Security mechanisms for REST service?
4. How can we specify the entry uri for a complete webservice application, where should it be specified (e.g.: - my application access uri is /login)
5. what is diff between POST and GET
6. Design REST endpoints for performing CRUD operations on customer accounts
7. URI for to get Single employee, All Employee , update Employee, Delete Employee  in that Employee is the class  REST Implementation for that.
8. Rest API disadvantages
9. Soap API message codes
10. How to develop REST webservice with spring.
11. Restful & API Architecture
12. Webservice as a publisher and as a consumer
13. What are the changes we need to do to convert MVC flow to Rest flow?
14. How is caching done in a web service

***Microservices:***

1. What is Microservices?
2. How to convert monolithic application to micro services

***Maven:***

1. Maven lifecycle and Jenkins
2. What is Jenkin?
3. 16. What did you do in Jenkin?

***Angular JS:***

1. Angular JS features and difference between angular JS 1 and AJS 2
2. What is typescript and features of angular 2
3. Services and registry mechanism in Angular JS
4. what are the ways to bootstrap angular application
5. How to notify changes in one controller to others in angular?
6. How to share variable between two controller in angular?

***Spring Boot:***

1. How to configure JBoss in SpringBoot
2. Spring boot security
3. Spring boot with REST and Hibernate.

***Java 8:***

1. Any java 8 features
2. Java 8 features(For each(Method code))
3. Why lambda expression introduced
4. Why do interfaces have default implementation methods in java 8
5. what is stream api
6. What is difference between interface with default methods & abstract class?

***Java 7:***

1. What are the features of Java 7?

***Multithreading:***

1. Multithreading questions for executor service and locking.
2. What is thread local? How it is used.
3. What is concurrency and how many concepts, you have used in concurrency?
4. Executer, ExecuterService, ThreadPoolExecuter, CountDownLatch, CycleBarier

***Junit:***

1. Junit questions for Mocking.
2. How to write a junit for private method.
3. Multi-threading concepts and where/how to use threading’s, usage of volatile

***Programs:***

1. Find the required sum of two numbers from given Array.
2. Once write this program calculate the time complexity of the program.
3. Another program to find the second highest number and time complexity.
4. Two classes one is base and another is derived class. And ask for which statements are correct.
   1. e.g.
   2. Base b = new Base ();
   3. Base b= new Derived();
   4. Derived d= new Base();
   5. Derived d= new Derived();
5. Write API or code for money transfer from one account to another.
   1. Requirement first authenticate the user, after successful transaction log all the detail for audit.
6. a)
   * 1. Employee e1 = new Employee();
     2. Employee e2 = new Employee();
     3. HashMap<Employee> hm = new HashMap<Employee>();
        + 1. hm.put(e1, "A");
          2. hm.put(e2, "B");
     4. System.out.println(hm.get(e1)+ " " +hm.get(e2));
        + 1. e2 = new Employee();
   1. System.out.println(hm.get(e2));
7. Singly link list program and find the loop.
8. Write hash map with own hash code and equal method.