**OOPS Concepts:**

1. Is-A vs Has-A relationship (difference)
2. Difference between Aggregation & Composition
3. What is Encapsulation? How does Lombok help here?
4. What is Abstraction?
5. Method Overloading vs Method Overriding (difference) (Compile time vs Runtime polymorphism)
6. Inheritance in java
7. Data Hiding concept
8. What is need and difference of abstract class & interface
9. Can we create object of Abstract class?
10. Can we override/overload Static method in java?
11. Can we declare “Abstract-static” combination for method? – is it valid?
12. Equals () and hashcode () method importance
13. Why we need to override equals and hashcode method in custom class?
14. Access Modifiers in java - What is default modifier/specifier for variable if we don’t provide any?
15. What approach Java uses pass by value or pass by reference?
16. What will be the output for following program?

Employee class has 2 fields (id and name)

Employee e1 = new Employee(1,"abc");

Employee e2 = new Employee(1,"abc")

What will be the output e1.equals(e2)? and why?

1. Transient and volatile keyword

**String:**

1. How many ways we can create String object? (diff & when to use which)
2. Importance of String Constant pool (SCP)
3. Why String is Immutable?
4. intern () method use w.r.to String class
5. Difference between String and String Buffer?
6. How to create our own Immutable class in java – write a sample program?
7. Difference between StringBuffer and StringBuilder?

**Collection:**

1. What is root interface in Collection framework?
2. ArrayList vs LinkedList (when to use which)
3. ArrayList internal working
4. How ArrayList grows in java?
5. HashSet internal implementation
6. HashMap internal working (changes in HashMap w.r.to Java 8)
7. HashMap vs Hashtable (when to use which)
8. ArrayList vs HashMap (when to use which)
9. TreeMap vs HashMap (whose performance is better)
10. TreeSet internal implementation.
11. TreeMap internal working
12. How we can sort Customized Objects with Map?

**Concurrent Collection:**

1. How ConcurrentHashMap works? (Reading and writing process)
2. Difference between HashMap & ConcurrentHashMap? (when to use which)
3. CopyOnWriteArrayList (usage and its advantages)

**Java 8:**

1. Lambda Expression (purpose/usage)
2. What is Functional Interface?
3. Different types of Functional Interface? (predicate, consumer etc.)
4. Difference between abstract class and interface after java8?
5. Default and static methods in java 8 (need and usage)
6. Dimond problem with default methods in multiple interfaces?
7. Stream Concept (internal working and advantages)
8. Stream vs Collection(difference)
9. Map vs FlatMap (difference)
10. How we can handle Exception in java8 / Optional class usage.
11. Parallel Stream (what is and when to use it and its advantages)
12. Using java 8 library, how we can remove duplicate elements from ArrayList of 100 elements.

**Exception:**

1. What is Exception?
2. What are types of exceptions
3. Difference between Exception and Error?
4. Exception hierarchy (root class for exception)
5. CheckedException vs UnCheckedException (difference)
6. Exception Handling – how to do it?
7. How to create our own Customized Exception?
8. Finally block Usage
9. Return statement w.r.to try, catch, finally block

**Threading:**

1. Ways of creating thread in java (which one is better and why)
2. Thread life cycle
3. Is it possible to get/obtain thread again which is in “Dead” state?
4. Callable vs Runnable (difference)
5. ExecutorService concept
6. Race Condition in java

**Serialization**:

1. What is Serialization? (process, use and advantage)
2. What is Deserialization?
3. In deserialization process, whether it will create new object or return same object reference?
4. Importance of SerialVersionUID in serialization process.
5. Transient keyword importance
6. Volatile keyword purpose

**Miscellaneous:**

1. What is cloning and its type and usages
2. Shallow cloning vs Deep Cloning
3. SOLID principles
4. Design pattern used in your project.
5. Singleton pattern / how to create custom singleton class ? what will happen if we clone singleton class object?
6. Comparator vs Comparable (differ and when to use which, syntax in java8)
7. Generics (usage, syntax and advantages)
8. Wrapper Class
9. What is Marker interface?
10. How to create our own Marker interface?
11. Inner class concept in java
12. Program to find Duplicate Element from Array (without collection library).
13. Program to find String occurrence from a File.
14. Program for Fibonacci series.
15. Program to convert integer number into Roman number.

**Spring:**

1. Spring Bean Life Cycle
2. IOC container concept in Spring
3. Application Context vs Bean factory (difference)
4. Dependency Injection in Spring
5. Types of Dependency Injection in spring (Setter vs Constructor - which one is better & when to use which)
6. Circular Dependency in Spring
7. Different types of Bean Scope in Spring.
8. Singleton scope vs Prototype Scope
9. How singleton scope in spring is different from java’s singleton object?
10. How we can inject Singleton bean into Prototype bean.
11. Autowiring in Spring (how many ways we can do it?)
12. @Autowired, @Qualifier annotation use
13. How many ways we can create Bean objects in spring?
14. @Bean annotation

**Spring MVC:**

1. Spring MVC Architecture flow (how it works internally?)
2. @Value annotation usage
3. @ConfigurationProperties annotation usage

**SpringBoot:**

1. Spring vs Spring Boot (difference)
2. SpringBoot features
3. Advantages of SpringBoot
4. Spring Boot Internal working
5. Bootstrapping in spring boot / How auto configuration works in spring boot?
6. @SpringBoot annotation usage (importance)
7. We don’t want to run spring boot application on tomcat server but on Jboss server, how we can do that?
8. How we configure two datasource (oracle and mongo dB) in SpringBoot?
9. In spring boot, can you deploy the changes without restarting the application?
10. Exception Handling in spring boot?
11. How we can call store procedure from Spring boot?

**Spring Data JPA:**

1. Difference between @Entity and @Table annotation

**Microservices:**

1. What is need of Microservices and its advantage?

**Hibernate:**

1. What is Hibernate and its advantages.
2. What is Dirty Checking in Hibernate
3. Object States in Hibernate
4. Fetching types (Eager vs Lazy loading)
5. Caching in hibernate (types)
6. Second level cache in hibernate (internal working)
7. How to generate primary key in hibernate?
8. Sequence generator in hibernate
9. Difference between Hibernate and JPA

**Webservice:**

1. SOAP vs REST (differences, which is better and when to use which)
2. Difference between PUT and POST?
3. Difference between @Controller and @RestController?
4. @RequestMapping annotation usage

**Docker**:

1. Different commands used in docker?

**Junit/Mockito:**

1. How we can test method which returns void?
2. How we can test static methods in junit/Mockito?
3. How we will mock static methods in power Mockito?
4. Mockito verify method

**Oracle:**

1. Function vs Stored Procedure (differences)
2. Primary key vs Unique key
3. Query to get nth (2nd,3rd etc.) highest salary from Employee table
4. Query optimization
5. Query execution plan
6. Write a query which will swap two column values.
7. Long tuning query optimization
8. Fastest way to determine if record exist
9. Row vs RowType difference in oracle
10. Why Cursors are used in Oracle
11. Triggers in oracle
12. Constraint concept
13. Cartesian / cross join
14. Table lock in database

**Agile:**

1. What are different types of process followed in Agile?