**Core Java**

**Note: Normal/core concepts, as per your choice.**

1. What is Interface what are the changes made after Java8?
2. Why String is immutable. With some practical thinking questions
3. What is Marker Interface. Why it is required if it has no data member and no member function.
4. Can Private and static method be override?
5. Create a immutable class. Write it on Notepad.
6. Create a singleton class. Write it on Notepad.
7. Is java pass by value / reference?
8. What are Equal and hascode contract?
9. What is Covariant return type?

**Collection API:**

1. **Difference between ArrayList and LinkedList in the java collection framework?**
2. **What are the features of Java Hashmap?**
3. **Differentiate between HashSet and HashMap**
4. **Explain internal working of HashMap. WRT to Java 8**
5. **What is the default size of the load factor in the hashing-based collection?**
6. **What is the difference between fail-fast and failsafe?**

**Java 8 :**

1. **What are the significant advantages of Java 8?**
2. **In Java 8, what is Method Reference?**
3. **What is an Optional class?**
4. **What is the lambda expression in Java and How does a lambda expression relate to a functional interface**
5. **What are static methods in Interfaces?**
6. **What is the default method, and why is it required?**
7. **Can a functional interface extend/inherit another interface?**

**Coding Based:**

1. Reverse a string using java 8
2. Remove duplicate from a list
3. Create a predicate and call the predicate
4. Given the list of integers, find the first element of the list using Stream functions?
5. Given a list of integers, find the total number of elements present in the list using Stream functions?
6. Given a list of integers, sort all the values present in it using Stream functions?

**Java 11:**

1. **Explain the var keyword in lambda parameters in Java 11.**
2. **Java 11 added the** [**List.of**](http://list.of/?trk=public_post-text)**() method. Can you explain its use and provide an example where it can be helpful over traditional methods of list creation?**
3. **How do you run a single-file Java program in Java 11?**

**Spring Boot**

1. What are the HTTP methods and their response codes?
2. How to handle exceptions in Spring Boot?
3. How to call REST services from another service?
4. What are Spring configurations?
5. What does the @SpringBootApplication annotation do internally?
6. What is the difference between @RestController and @Controller in Spring Boot?
7. Given a class Book.Class Create a controller to demonstrate any one of the methods
   1. Get a book detail by its ID
   2. Add a book detail
   3. Delete a book detail by its ID

**Microservices**

1. How do you validate your requests?
2. How does one microservice call another microservice?
3. What is a profile? How to set profiles for different environments?
4. What is fallback? How does the Circuit Breaker design pattern work?
5. What is Hystrix?
6. What are deployment strategies? How to deploy code to production without impacting live consumers?
7. What is rate limiting and throttling?

Scenario based:

* 1. One service becomes unresponsive and delays responses
  2. How will you Ensure secure communication between microservices
  3. How will secure internal service-to-service communication and external user access.
  4. How will you update dependent services without causing downtime\
  5. How will you handle one service which is depends on another, which is temporarily down?

**DevOps:**

**Docker:**

1. What is a Container?
2. Can you tell something about docker namespace?
3. What is the docker command that lists the status of all docker containers?
4. What is the purpose of the volume parameter in a docker run command?
5. Describe the lifecycle of Docker Container?
6. What command is used for remove all stopped containers, unused networks, build caches, and dangling images?
7. What command is used for viewing all running containers?
8. What command is used to go inside a running container?

**Kubernetes**

**1**. Explain Kubernetes Architecture.

2. What is a Pod in Kubernetes?

3. How does Kubernetes handle container scaling?

4. What is Kubelet?

5. What is a Service in Kubernetes?

6.  Explain the concept of Ingress in Kubernetes.

7. What is a Namespace in Kubernetes?

8. How can you achieve communication between Pods in different Nodes?

9. In Kubernetes, what are the various services available?

**AWS**

1. What is a secret manager?
2. What is an API gateway?
3. What is a Lambda service?
4. What are Step Functions?
5. How do you secure data in S3?
6. What is AWS CloudFormation, and how does it facilitate DevOps practices?