Lead Java Developer I Assessment

1. Name *

Enter your answer

2. Email ID *

Enter your answer

3. Location *

Enter your answer

4. Notice Period *

Enter your answer

5. CTC *

Enter your answer

6. ECTC *

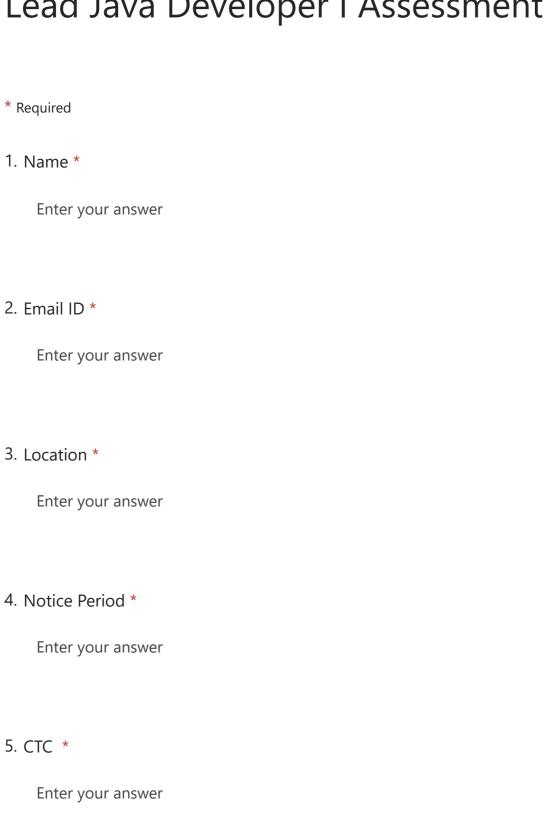Enter your answer

7. Experience in years *

The value must be a number

8. Mobile number *

The value must be a number

9. In a microservices architecture, which pattern helps to reduce the coupling between services by ensuring that each service manages its own data storage? (10 Points) *

○ Gateway Pattern

○ Service Registry Pattern

◉ Database Sharding Pattern

○ Data Partitioning Pattern

10. Which pattern is used to ensure eventual consistency between microservices when multiple services need to update their data independently? (10 Points) *

○ Event Sourcing Pattern

◉ Saga Pattern

○ CQRS Pattern

○ Retry Pattern

11. Which pattern helps to ensure message delivery and communication reliability between microservices by allowing messages to be retried? (10 Points) *

○ Saga Pattern

○ Retry Pattern

● Circuit Breaker Pattern

○ Event Sourcing Pattern

12. What is default scope of bean in Spring framework? (10 Points) *

● singleton

○ prototype

○ request

○ session

13. Which of the following annotations is used to define the main class of a Spring Boot application? (10 Points) *

○ @SpringBoot

○ @SpringApplication

○ @SpringConfiguration

● @SpringBootApplication

14. Which of the following is a feature provided by Spring Boot Actuator? (10 Points) *

○ Security

○   Monitoring and managing application health

○   Dependency injection

○   Database access

15. Which file should be modified to change the packaging of a Spring Boot application from JAR to WAR? (10 Points) *

◉   pom.xml

○   application.properties

○   build.gradle

○   application.yml

16. Which configuration is required to deploy a Spring Boot WAR file to Apache Tomcat? (10 Points) *

◉   Modify the Tomcat server.xml file

○   Enable the Spring Boot auto-configuration property

○   Configure the application.properties file to specify the deployment type

○   Configure the pom.xml file to include the necessary dependencies

17. Can you deploy multiple Spring Boot applications to the same Apache Tomcat instance? (10 Points) *

◉   Yes, but each application must have a unique context path

○   No, Apache Tomcat only supports one Spring Boot application per instance

○   Yes, Spring Boot handles application isolation automatically

○ No, Spring Boot applications cannot be deployed to Apache Tomcat

18. How can you enable SSL (HTTPS) in a Spring Boot application deployed on Apache Tomcat? (10 Points) *

○ Configure SSL settings in the [application.properties](#) file

○ Add SSL certificates to the Tomcat keystore

● Set up SSL using Apache HTTP Server as a reverse proxy

○ Enable SSL using annotations in the Spring Boot application code

19. How can you configure Spring Security to use a JWT request filter in a Spring Boot application? (10 Points) *

○ By adding a bean definition in the application context

○ By annotating the filter with @Component

● By registering the filter in the security configuration class

○ By configuring it in the [application.properties](#) file

20. When should a JWT request filter be applied in the Spring Boot request processing flow? (10 Points) *

○ After authentication

● Before authentication

○ After authorization

○ Before authorization

21. What is the role of encryption keys in encryption algorithms? (10 Points) *

○ To compress data before encryption

○ To generate random numbers

● To encode plaintext into ciphertext

○ To control the encryption and decryption processes

22. Which pattern is used to optimize the performance and scalability of microservices by distributing and balancing the workload across multiple instances? (10 Points) *

● Load Balancer Pattern

○ Observer Pattern

○ Singleton Pattern

○ Circuit Breaker Pattern

23. What is true about <list> collection configuration elements? (10 Points) *

○ This helps in wiring a list of values, allowing duplicates.

○ This helps in wiring a list of values but without any duplicates.

● This can be used to inject a collection of name-value pairs where name and value can be of any type.

○ This can be used to inject a collection of name-value pairs where the name and value are both Strings.

24. What is bean autowiring? (10 Points) *

○ Autowiring lets Spring resolve collaborators otherbeans for your bean by inspecting the contents of the BeanFactory without using <constructor-arg> and <property> elements.

○ Autowiring injects values in spring beans.

● Autowiring injects one bean into another.

○ Autowiring helps in wiring a list of values, allowing duplicates.

25. What is byType mode of autowiring? (10 Points) *

○ Default setting which means no auto-wiring and you should use explicit bean reference for wiring.

○ Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.

● Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.

○ Autowiring by property type. Spring tries to match and wire a property if its type matches with exactly one of the beans name in configuration file.

26. What does @Required annotation mean?
(10 Points) *

○ This annotation indicates that bean property must be populated by the user.

● This annotation indicates that bean property is required while saving the bean data to database.

○ This annotation simply indicates that the affected bean property must be populated at configuration time, through an explicit property value in a bean definition or through autowiring.

○ This annotation indicates that bean property is required while serializing the bean.

27. What is RequestHandledEvent:event?
    (10 Points) *

    ○ This event is published when the Servlet Context is either initialized or refreshed

    ○ This event is published when the HTTP Request is received.

    ○ This event is published when the HTTP session is initialized or refreshed.

    ● This event is published when the HTTP Request is serviced.

28. What is circuit breaker pattern, and how does it relate to microservices?
    (10 Points) *

    ○ Circuit breaker pattern is used to manage electrical circuits in microservices
      architecture

    ● Circuit breaker pattern is a design pattern to handle network failures and latency in
      distributed systems

    ○ Circuit breaker pattern is used to enforce rate limiting in microservices

    ○ Circuit breaker pattern is a technique for load balancing across microservices

29. What are some common challenges in microservices architecture? (10 Points) *

    ● Service discovery, data consistency, and inter-service communication

    ○ Monolithic deployment, centralized management, and synchronous communication

    ○ Tight coupling, shared databases, and distributed transactions

    ○ All of the above

30. What is the purpose of Spring Cloud Netflix Eureka in microservices
    architecture? (10 Points) *

○ Service registration and discovery

○ Load balancing

○ Inter-service communication

○ Centralized configuration management

31. What is the role of Spring Cloud Config Server in microservices architecture? (10 Points) *

○ Service registration and discoveryOption 1

◉ Centralized configuration management

○ Implementing circuit breakers

○ API gateway

32. What is the primary purpose of the Spring Cloud Security component in microservices architecture? (10 Points) *

○ Implementing service discovery

○ Managing API gateways

○ Configuring distributed tracing

◉ Providing security features for microservices

33. Which of the following is a benefit of using HikariCP in a Spring Boot application? (10 Points) *

○ Increased memory usage

○ Reduced database performance

○ Better performance and lower resource consumption ⦿

○ Increased network latency

34. Which configuration property can be used to customize the HikariCP connection pool size in application.properties in Spring Boot? (10 Points) *

○ hikaricp.connectionTimeout

○ hikaricp.maximumPoolSize

○ hikaricp.minIdle

⦿ hikaricp.idleTimeout

35. What happens if the maximumPoolSize is reached in HikariCP? (10 Points) *

○ New connections are queued indefinitely

○ The application crashes

○ The oldest connections are closed to make room for new ones

⦿ The connection pool size is automatically increased

36. Which of the following is a recommended best practice when using HikariCP in a Spring Boot application? (10 Points) *

○ Setting the maximumPoolSize to a very low value

○ Configuring a long connectionTimeout value

⦿ Tuning the connection pool size based on application requirements

○ Disabling idle connection tracking

Microsoft 365