|  |
| --- |
| 1. You are developing a web application that requires a centralized mechanism to configure and inject dependencies. Question: Which Spring feature allows you to centralize dependency injection and configuration across your application?    a) Spring MVC   b) Spring Boot   c) Spring Bean   d) Spring IoC Container |
| 1. You need to perform a task in parallel using Java 8 streams on a large list of data. Question: Which Java 8 feature would you use to achieve parallel execution?    a) Stream.map()   b) Stream.parallelStream()   c) Stream.forEach()   d) CompletableFuture.runAsync() |
| 1. You are designing an application where the creation of complex objects needs to be simplified and controlled. Question: Which design pattern is most suitable for this scenario?    a) Singleton   b) Builder   c) Prototype   d) Factory |
| 1. You want to ensure that a particular Spring bean is created only once and shared across your application. Question: Which bean scope should you use in Spring?    a) Prototype   b) Request   c) Singleton   d) Session |
| 1. You are required to apply different operations on a list of items conditionally based on the presence of values. Question: Which Java 8 feature allows you to apply conditional operations on values?    a) Optional.ifPresent()   b) Optional.map()   c) Stream.filter()   d) Stream.collect() |
| 1. You are building a multi-tiered application where the service layer needs to be loosely coupled from the data access layer. Question: Which design pattern is most appropriate for decoupling these layers?    a) Observer   b) Adapter   c) Strategy   d) DAO (Data Access Object) |
| 1. n a Spring-based application, you need to select a particular bean implementation dynamically at runtime based on some condition. Question: Which Spring feature allows you to achieve this?    a) @Autowired   b) @Qualifier   c) @Primary   d) @Profile |
| 1. You are implementing a system where certain operations need to be deferred until they are needed, which also allows you to change the method of execution at runtime. Question: Which design pattern is best suited for this requirement?    a) Command   b) Proxy   c) Decorator   d) Chain of Responsibility |
| 1. You are optimizing a real-time data processing pipeline and want to utilize multi-core processing capabilities in Java 8. Question: Which feature or pattern would you use to optimize CPU utilization?    a) Fork/Join framework   b) CompletableFuture   c) Stream API   d) ExecutorService |
| 1. In a large-scale application, you need to ensure that certain operations, such as sending notifications, are performed in the background without blocking the main thread. Question: Which Java 8 feature allows you to handle such asynchronous tasks?    a) Stream.reduce()   b) CompletableFuture   c) Optional.orElse()   d) Lambda Expressions |

|  |
| --- |
| 1. Difference between Spring and Spring Boot |
| 1. Spring Boot auto-configuration/disable auto-configuration |
| 1. Spring JPA-Query pagination |
| 1. How to call a stored procedure using Spring JPA |
| 1. Spring Security/JWT generation/validation |
| 1. Spring IoC/Bean Scopes |
| 1. What may be the problem if you add too many logs |
| 1. What needs to be handled in complete API call coding? (like unit testing, validation, exception handling) |
| 1. How to update properties without redeploying the application? |
| 1. If you are facing a memory error, how would you handle it? |