1. \*\***Singleton Keyword:\*\* It**  is a design pattern used in object-oriented programming to ensure that a class has only one instance and provides a global point of access to that instance.

2. **\*\*Make a Singleton Class:\*\*** To create a singleton class, you typically make the constructor private to prevent direct instantiation, provide a static method to access the single instance, and use lazy initialization or eager initialization based on your requirements.

4. **\*\*JOIN Keyword in SQL:\*\*** The JOIN keyword is used in SQL to combine rows from two or more tables based on a related column between them. Common types of joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

5. **\*\*Primary and Foreign Key Concept in SQL:\*\***

primary key uniquely identifies a record in a table, And foreign key establishes a link between two tables.

6. \*\***Composite Key in Relational Database:\*\***

**It** is a key that consists of two or more columns, used together to uniquely identify a record in a table. It is used when a single column cannot guarantee uniqueness.

7. **\*\*Spring Initialization:\*\***

Spring framework provides various mechanisms for initializing beans, such as constructor injection, setter injection, and autowiring. Beans are typically defined in Java configuration files.

8. **\*\*IOC (Inversion of Control):\*\***

IOC is a design where the control of object creation and dependency injection is shifted from the application code to a container. It helps achieve loose coupling and better manageability.

9**. \*\*How IOC Container Works in Spring:\*\***

Spring IOC container manages the lifecycle of beans, their creation, configuration, and injection of dependencies. It uses configuration metadata (XML, annotations, or Java configuration) to wire beans together.

10. **\*\*Advantages of Spring over Java:\*\***

Spring provides features like dependency injection, aspect-oriented programming, unit testing, and modularity.

11. **\*\*Lambda in Java 8:\*\***

Lambda expressions are anonymous functions introduced in Java 8. They allow you to write expressive code, especially when working with functional interfaces.

12**. \*\*Reason for Lambda Introduction in Java 8:\*\*** Lambda expressions were introduced to simplify the use of functional interfaces and enable functional programming constructs in Java.

13. **\*\*Spring Jobs:\*\***

Spring can handle various job-related tasks through features like Spring Batch and scheduling using Spring's built-in mechanisms or third-party libraries.

14**. \*\*Handling Load on an API:\*\***

It is like load balancing, caching, optimizing database queries, and employing scalable

22. **\*\*Java Patterns:\*\***

Design patterns are reusable solutions to common problems in software design. like Singleton, Factory, Observer, and Strategy patterns.

23. **\*\*Spring Boot Advantages:\*\***

Spring Boot simplifies the process of building and deploying Spring applications by providing auto-configuration, embedded servers, and production-ready features.

27. **\*\*RestController vs. Controller:\*\***

**`@Controller**` is used to create a controller class that handles web requests and returns views

**`@RestController**` combines `@Controller` and `@ResponseBody`, returning data in a format like JSON.

28. **\*\*Response Body Location:\*\***

the `@ResponseBody` annotation indicates that the return value of a method should be serialized and sent in the response body.

31. **\*\*FlatMap():\*\*** `

is a method in Java's Stream API that's used to compress a stream of streams into a single stream. It's used in scenarios where you have nested collections.

32. **\*\*Functional Interface:\*\***

functional interface is an interface that contains only one abstract method. It is used as a basis for lambda expressions

33. **\*\*Bean Scope:\*\***

a bean scope defines the lifecycle and visibility of a bean. Common scopes include singleton (one instance per container), prototype (new instance per request)

34. **\*\*Property of Tree Map:\*\***

A TreeMap is a sorted map implementation in Java that stores its entries in a sorted order based on their keys.