1. Why we are passing String[] arg in main method ?

Ans: to compile the program in command

1. What happens if we have main method as protected ?

Ans: JVM won’t have the access the program will not run or compile.

[#SpringBoot](https://www.linkedin.com/feed/hashtag/?keywords=springboot&highlightedUpdateUrns=urn:li:activity:7229874783201468418)

1. What are the different bean scope available in spring boot? what is the default one ?

2. What are the different feature provided by spring boot?

3. How do you connect to data base from spring boot application?

4. How does [application.properties](http://application.properties/) file do its job internally?

5. How do you deploy your project ?

6. What is no Sql data base ?

7. Have you created any custom bean ? How do you create bean?

8. How does authentication and authorization works in spring boot?

9. How do you check logs?

10. What is spring boot? what the advantages of spring boot.

11. what is actuator and what are the feature of actuator?

12. Difference between .yml file and .properties file

13. What is IOC container?

14. What is dependency injection?

15. What are different annotations available in spring boot?

16. What is the use of properties file in java/spring boot

17. What are different ways to create spring boot project.

18. Does spring provide thread safety for spring bean?

Spring vs spring boot /starter web /boiler plate tea are reduce in spring boot

If tomcat server is embedded with spring boot.

-> Alternate  way we need create  artefact as war file and then take app container and  deploy in jetty server

Profiles in Spring boot.

[#Java](https://www.linkedin.com/feed/hashtag/?keywords=java&highlightedUpdateUrns=urn:li:activity:7229147269294428160):

1. What is concurrent modification exception?

2. Which interface in collection framework allows us to remove duplicate.

3. Can I have 2 null values in hashMap?

4. Difference between list-iterator and iterator?

5. Can we have a try block without having catch and finally?

6. If I write return / System.exit() statement inside try block will finally block gets executed?

1. What is the significance of static keyword in java?

2. What is fail fast?

3. What is the feature available in java 8?

3. What is parallel stream in java?

4. Explain internal implementation of set in java.

1.MethodOverloading vs MethodOverriding

2.Interface vs Abstract class

3.Exceptions /types/try/catch/finally

4.HashMap vs HashTable

5.ArrayList And LinkedList

6.== vs equals

7.Immutable in java?

-> Declare the class as final

-> **all fields private and final**:(Emp Class)

-> create a parameterized constructor to all fields

-> **Do not provide setters**:

-> **Provide only getters (no setters)**:

-> **Thread Safety**: No need for synchronization as the state can't change.

1. Oops
2. Wrapper class?
3. Inheritance?
4. Exception
5. Try catch ?
6. Garbage collection | Daemon thread
7. linked | array list
8. Create an immutable class?
9. How to pass Obj as key in hashMap ?
10. Which is best Search algorithm/bubble sort?  
    -> **For searching**: Binary search is much faster than linear search if the data is sorted.Time complexity : O(log n) .  
    -> **For sorting**: Bubble sort is not recommended for large datasets due to its poor time complexity. Instead, faster sorting algorithms like **merge sort** or **quick sort** are often preferred for large data.Time complexity : O(n2) (worst),O(n)(best)
11. Design pattern in Corejava? singleton?/ cloenable interface to break the singleton
12. Solid principles/failFast/failSlow?
13. Default methods in java8
14. Java 11 features
15. Sql injection for security
16. Joins/ inner join /outer join /
17. Sql query /2ndLargest/update/
18. RestAPI
19. Stream and parallel stream in java8
20. Static variables are class-level variables, so their values are not part of the object's state. Hence, they are not included in the serialization process.
21. hashSet internal working ?/ internally they using the hasp map only.
22. How we can make the HashMap as synchronise / Collection class  inside that there method called synchronised method used to hashMap as synchronised hash map

23.Micro Service  design pattern ?

1. @spring actuator || admin server  /@spring profile /@transtion
2. Controller vs rest controller
3. Micro Service communication? / syn(REST Tem) & async(Kafka.,)

————

Java8 program /oddEven/occ/

Pattern program

===========

1. Java8 coding
2. Hash Collision
3. String pool
4. Heap vs stack
5. Concurrent map
6. Exceptions/inner class
7. Interface vs abstract
8. Use of abstract class
9. Multi threading
10. Request mapping /put/post/patch
11. Joins

======== JI-1 =============

List l=[1,0,2,0,3,0,4,0,5,0];

output : 1,2,3,4,5,0,0,0,0,0

String S= "Hello";

s.concat("World");

S.O.P(S);

 String S1=s.concat("World");

S.O.P(S1)

====================

 -> sec largest no in array; // with duplicate

-> String occurrence

-> student : iterate the student details

-> actuator ??

-> @springboot? @Bean where why? @controller ?

-> @component vs @bean class/method

-> @profile

->@controller @getMapping?REST API

-> @Rest controller vs Controller

-> @Autowire /@service

-> JPA /crud

-> inner join

-> SQL Query in depth ??/student db

-> query to find the sec largest salary in the employ table?

-> Jetty Tomcat?

-> micro service

Builder design pattern ? Factory design pattern ?

Do you know try-with-resources blocks? Why do we use them? When are they introduced?

=========================

Volatile / Transient

Memory module

HTTP request?

Authencation vs authrization

Fail fast vs fail safe

Create db in code

Filter / reduce in streams

———

Experience with IoC containers and DI frameworks (like Guice), SOLID and DRY principles, REST frameworks,

—

Mapping?

DML/DDL

How to get first 10 rows in from the table which have 10k rows for pagination;

Delete/drop/truncate

Trigger / Index /

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Unit Testing

Kafka

SpringSecurity

Ansible   
terraform   
AWS / Azure

DSA

Git Stash

Git Fork

Git Time Travel

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*