

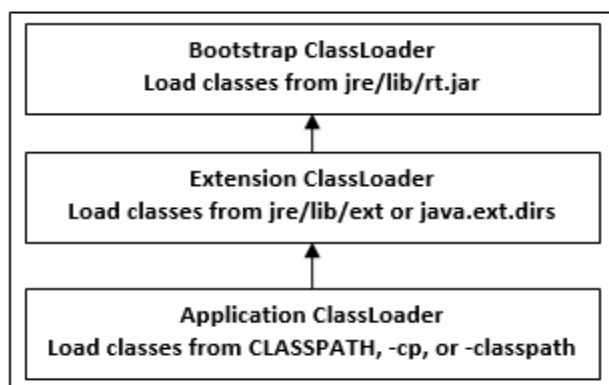
1 JAVA

Basic Core Java

1) Tell me about class loaders?

Java ClassLoader is an abstract class. It belongs to a **java.lang** package. It loads classes from different resources. Java ClassLoader is used to load the classes at run time. In other words, JVM performs the linking process at runtime. Classes are loaded into the JVM according to need. If a loaded class depends on another class, that class is loaded as well. When we request to load a class, it delegates the class to its parent. In this way, uniqueness is maintained in the runtime environment. It is essential to execute a Java program.

2) Types of class Loaders



- 3) How do you achieve loose coupling and high cohesion in Java?
- 4) What are the methods present in Object Class?
- 5) Explain JVM, JDK and JRE.
- 6) How does jvm work internally ?
- 7) Can we create objects for abstract class?
- 8) How to make an immutable class with mutable references like List.
- 9) Tell me a way where I can create an object without using a constructor. What security practices to be followed while creating a copy constructor.
- 10) Part-of vs Has-a relationship.
- 11) Difference between String and StringBuffer and StringBuilder?
- 12) Why String buffer use
- 13) final and finally and finalize
- 14) Difference between abstract class and default methods?
- 15) transient
- 16) Volatile
- 17) diff instance variable and local variable
- 18) Static, final. Why static ?
- 19) Can we create a private constructor? if yes, purpose?
- 20) Can you overload the main method ?
- 21) What is the interface ?
- 22) What is immutable in java , how to define an immutable class?
- 23) What is composition in java ?
- 24) Serialization, Reflections, Generics, Strings(In depth).
- 25) Difference between aggregation/ association.

- 26) In how many ways can we create an object in JAVA?
- 27) Why method overriding is not applicable for static methods?
- 28) Why are multiple inheritances not supported apart from being ambiguous in nature.?
- 29) Marker interface. What does it do?
- 30) What is association, aggregation and composition?
- 31) wap to show association, aggregation and composition?
- 32) Java 7 features?
- 33) create your own immutable class?
- 34) hashmap internal work?
- 35) abstraction vs interface?

OOPS

- 1) OOps Concept

Memory management

- 1) Heap memory partitions?
- 2) garbage collection

Collections

- 1) Hashmap internal structure?
- 2) What if we don't override the hashCode method in hashmap?
- 3) Difference between Map and FlatMap ?
- 4) What changes have been made in Hashmap in java 8?
- 5) What is ConcurrentModificationException?
- 6) What is Fail Safe and Fail Fast?
- 7) How does ConcurrentHashMap work?
- 8) How to use synchronised hashmap?
- 9) What is the difference between Comparable and Comparator ?
- 10) What feature java 8 has given for Comparator?
- 11) HashMap vs HashSet.
- 12) hashmap and treemap
- 13) Create your own priority queue without using inbuilt DS. Why do we need a comparator in the priority queue?
- 14) Which all thread safe collections in java?
- 15) Difference between list and set
- 16) Diff between comparable & comparator?
- 17) What feature java 8 has given for Comparator?
- 18) Different blw map & flatMap
- 19) What is an ordered collection and sorted collection?
- 20) What is all about hashCode and equals?
- 21) memory overflow of array list?
- 22) If an employee object is stored in a hashmap. How is it stored?
- 23) diff b/w arraylist and linkedlist?
- 24) wap to add employee objects in arraylist?
- 25) wap to add data into arraylist?
- 26) reverse arraylist using collection?
- 27) reverse arraylist using comparator?
- 28) find all elements from arraylist using lambda expression?

29)

Multithreading

- 1) Ways to create Thread in java?
Creating a thread. There are two ways to create a thread in java. First one is by extending the Thread class and second one is by implementing the Runnable ...
- 2) can we call start 2 times
- 3) demon thread
- 4) Life Cycle of thread.
- 5) Synchronized block purpose
- 6) Why wait, notify and notifyAll methods present in Object Class instead of Thread Class?
- 7) What is the difference between callable and runnable interface?
- 8) What is Future in java?
- 9) Best practices to be followed for multithreading. Reentrant lock? Why not use them very often?
- 10) Multithreading- diff bw Run and start, what happens if run is executed for thread inspite of start
- 11) What if you create two threads in a system which has only a single core processor?
- 12) How to create thread without using Runnable or Thread?
- 13) What are the benefits of parallel processing? Tell me some scenarios when we can & can't use parallel processing.
- 14) Volatile vs synchronised?
- 15) What is ThreadLocalMap? Why should we use that?
- 16) Why wait, notify and notify all methods in object not in thread?
- 17) Difference between sleep and wait?

Serialization

- 1) What is Serialisable?
- 2) static serializable or not?
Static variables belong to a class and not to any individual instance. The concept of serialization is concerned with the object's current state ..

Exceptions

- 1) Exception and error definition.
- 2) Checked vs unchecked Exceptions?
- 3) how to write custom exception
- 4) Explain catch with parent exception and another catch with child exception.
- 5) Exception handling. Create your own custom exception. Why chain of responsibility design pattern is suggested to be followed during exception handling.
- 6) Difference between throw and throws?
- 7) Exceptions order in catch block?
- 8) try block with resources and how are those resources closed?
- 9) difference between class not found exception and class def exception?

Java 8

1. What is the default method in java8?
2. Can I write another method inside a default method?

3. functional interface and stream connection.
4. What are the java 8 features ?
5. What are the advantages of stream apis?
6. You have a collection of employees, write logic to extract employee names using java 8?
7. What is Predicate, supplier, consumer in java?
8. What is the difference between Collections and Stream ?
9. Why is permgen space removed from JAVA 8 ? Why was metaspace introduced in JAVA 8 ?

Spring

1. How do you do some configuration before the spring bean container starts?
2. Why do we use PostConstruct in spring?
@PostConstruct is an annotation used on a method that needs to be executed after dependency injection is done to perform any initialization.
3. Different modules of spring that you have worked on.
4. Dependency Injection. Explain types. Which one is better?
Constructor injection — good, reliable and immutable, inject via one of the constructors. Possible to configure in: XML, XML+Annotations, Java, ...
5. Difference between Singleton and session prototype.
Prototype scope = A new object is created each time it is injected/looked up. It will use new SomeClass() each time. · Only one instance will be created for a single bean definition per Spring IoC container and the same object will be shared for each ..
6. Explain about spring application security
7. How do you call an api in your app and validate it?
Call arguments should be passed as JSON in the body of the request with content-type HTTP header set to application/json.
8. How did you handle db errors in your application? Logs slf4j
9. How do you call a db query in your spring app?
if we create a query method called findById() and annotate it with the @Query annotation, Spring Data JPA won't ...
10. Can you only download dependencies from internet or u can get in your local by sharing
11. Is it possible to create a private repository ?
12. Questions about Swagger and Yml
13. What is partial dependency in Spring?
Circular dependencies or partial dependencies result with Setter DI because object creation happens before the injections.
14. How to do injection in Spring?
The injection in Spring is either done via setter injection or via construction injection.
15. what is composite key, how retrieve and how to add primary keys.
Composite keys are table properties that use two columns as the primary keys.
16. What happens when we create an inner bean of prototype scope in a singleton scope bean?
17. Configure bean dependencies as much as possible, try to avoid autowiring as much as possible. Why should avoid auto wiring please elaborate more on this.
18. Ans: You should always avoid autowiring by type/name since it is lookup-based and can be expensive in a large enterprise-level application. Also if you use @Autowire it

makes testing difficult in frameworks like Mockito. Try to use @Autowire by constructor level.

19. Write a code for implementing inheritance in the Spring
20. ANs>implementing inheritance in spring is inheriting the common properties of beans using , this will inherit all the properties of the parent bean into the new bean.
21. In which case @Autowired is not preferable?
22. Ans>@Autowired does not allow the programmer to enable the restriction on specific property. it allow the programmer to enable restriction on simple, object, all
23. In which scenario, we have to prefer Spring-JDBC rather than Spring-Hibernate?
24. .What happens if we autowire all the objects?
25. Would you post about the difference between Abstract factory and Spring DI?
26. when to use constructor injection and when to use setter injection with different scenarios?
27. "Types of Dependency Injection. Which one is more efficient and why?"
28. I've been asked, among the rest, about @Qualifier and @Autowired annotation.
29. How to connect multiple Databases with spring
30. Can Spring Controller instance's method return the below:
 - 1) null
 - 2) SpEL
 - 3) com.u2ware.springfield.view.jstl.JstlViewResolver
31. What is default scope of spring bean
32. Spring bean scope?

springBoot

1. Why spring boot?
2. How to change tomcat's default port ?
3. How to get property from Yml file?
YAML files can't be loaded via the @PropertySource annotation. So in the case that you need to load values that way, you need to use a properties file.
4. Most used annotations and its explanation
5. Connection api used for connecting to DB?

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/bookshop
spring.datasource.username=root
spring.datasource.password=password
@Autowired
private JdbcTemplate jdbcTemplate;
```

6. In JPA how to create a connection to oracle or mySQL?

4. Connect to MySQL with Spring Data JPA

Spring Data JPA provides more advanced API that greatly simplifies database programming based on Java Persistence API (JPA) specification with Hibernate as the implementation framework.

You need to declare dependency for Spring Data JPA as follows:

```
<dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>
```

For data source properties, before the URL, username and password you can also specify these additional properties:

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format_sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL
L57InnoDBDialect
```

And then, you need to code an entity class that maps to a table in the database, for example:

```
import javax.persistence.*;

@Entity

@Table(name = "users")

public class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Integer id;


    private String email;


    private String password;


    // getters and setters...

}
```

And declare a corresponding repository interface:

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface UserRepository extends JpaRepository<User,
Integer> {
```

```
}
```

And then you can use the repository in a Spring MVC controller or business class like this:

```
@Controller
```

```
public class UserController {
```

```
    @Autowired
```

```
    private UserRepository rep
```

```
    @GetMapping("/users")
```

```
    public String listAll(Model model) {
```

```
        List<User> listUsers = repo.findAll();
```

```
        model.addAttribute("listUsers", listUsers);
```

```
        return "users"
```

7. Any SQL db used?

8. What are conditional beans in spring boot

Spring Boot includes a number of @Conditional annotations that you can reuse in your own code by annotating @Configuration classes or individual

9. Dependencies in Spring Boot--how they are described in pom

10. How do you handle Exceptions in your application?

11. How to configure JPA in spring boot?

5. Spring Data JPA Repository Configuration ... To activate the Spring JPA repository support, we can use the @EnableJpaRepositories annotation ..

12. How are you executing spring boot application?

13. How does autoconfiguration help in spring boot?

14. Explain @SpringBootApplication.-

@componentscan @configuration @EnableAutoConfiguration

15. Spring Boot components - like @Bean , @SpringBootApplication or

@SpringbootConfiguration ...which is better to use ?(this helps to minimize /exclude /disable those beans which are auto configured through bean annotation) , ConditionalOnProperties in Spring

16. Which type of property file is better(Yml or application.properties)?

17. How are you maintaining security in your application ? In case of partial dependency we will use Constructor or setter based injection?

18. How to deploy spring boot application in external server by disabling embedded server/ how to.
Use `@SpringBootApplication(exclude = {EmbeddedServletContainerAutoConfiguration.class, WebMvcAutoConfiguration.class})`.
19. Difference between `@Repository` and `@Service` annotations?
20. If the interviewer asks what all Spring boot annotations have you used then what will be the answer?
21. Ans -> Mostly tell them the common annotations like, `@component` `@service` `@repository` `@controller` `@bean` `@getmapping` `@postmapping` `@resquestBody` `@springbootapplication` `@EnbaleConfiguration` `@ComponentScan` `@Configuration` etc etc Many other annotations are also there
- 22.

Microservices

1. Advantages of Microservices?
2. Challenges in microservices ?
3. Solutions to the challenges?
4. Difference bw Monolithic vs Microservices
5. Microservice Architecture
6. How are you communicating between microservices?
7. APIs required to call a third party api from your service.
8. Do you know about Feign Client? Explain
9. Feign as RestClient...
7. Declarative REST Client: Feign ... Feign is a declarative web service client. It makes writing web service clients easier. To use Feign create an interfac
What Is a Feign Client? ... Netflix provides Feign as an abstraction over REST-based calls, by which microservices can communicate with
10. Any knowledge on springCloudConfigServer? Explain
11. Any Load balancer you have used?
12. Communication between microservices
13. In microservices how github plays its role?
14. What is Circuit Breaker Pattern? API Gateway ? SAGA Pattern ?
Use the saga design pattern to ensure data consistency on distributed transactions in microservices architectures.
A saga pattern is a sequence of local transactions where each transaction updates data within a single service. The first transaction in a .
15. How to track/log the requests which go through many microservices ?

Spring MVC

How to handle sessions in Spring MVC?

Spring Session provides an API and implementations for managing a user's session information while also making it trivial to support ...

... different ways to store attributes in a session with Spring MVC. ... treat our TodoList as session-scoped by using `@SessionAttributes`:

how to configure two databases in spring MVC?

Use Multiple DataSources in Spring MVC · 1- The objective of the document · 2- Preparing the Database · 3- Create Maven Web App Project · 4- Configure

Role of handler mapper, how is the url mapped to a particular controller in their method?

If we have an interface and it has two implementation classes in a Spring MVC project, then what will happen or rather which class will load first when we start the server ? Given , @Qualifier is not mapped to any of the implementation classes.

What happens when you mention the view name as empty or NULL in your controller ? How will it behave ?

Where do we define the name of FrontEndController.java class in web.xml as dispatcherServlet? Means how Tomcat knows this FrontEndController.java class is servlet class .

If you have 1000 fields on the JSP page then how you will get all the fields in spring controller without using @ModelAttribute("").

Ans: We can use @RequestParam Map<String, String> in the place of @ModelAttribute("")

Can we have multiple web.xml in an application?

Ans>Technically YES !! Only web.xml file inside WEB-INF will be used as deployment descriptor. Rest all files can be used as normal xml files.

What if more than one Controller and its method have the same URL Request mapping method, How does DispatcherServlet take calls?

What is the difference between postHandle() and afterCompletion()?

Can we say Interceptors are Spring MVC filters? If not, what is the difference?

how you will call one controller method from another controller method in spring MVC, and how you will declare a controller method from which all the requests will pass through and contact another controller method for further request processing?

Ans: I find no reason for calling a controller's method from another controller (except it's URL redirect). Controllers are not meant to have any business logic, they should only receive the HTTP request, validate it and pass it to the manager for handling. In any scenario where you are required to call another controller's method – is flawed code/design – and personally I never came across such scenario in my career.

What are the different types of view resolvers ,can you please explain them.

Can we have empty requestMapping declaration

```
@RequestMapping()  
public void defaultMethod(){  
    System.out.println("I am Sachin");  
}
```

Can we call one method that has requestMapping annotation from another method that also has some requestMapping annotation?

How can we create a default method in spring MVC that always executes?

What is dif between @RequestParam and @PathVariable

What is difference between URI and URL

Docker

1. What are the steps in deployment in your application?
2. Docker command to go inside the container?
3. Docker Image?
4. What is container orchestration and why should we use it?
5. How to use docker in microservices

MongoDB

1. How are you writing queries in MongoDB?
2. How to create and drop collections in MongoDB?
3. Method name to create collections in MongoDB?
4. What is shredding?

Shredding allows you to get the “bad” stuff over with in one go ... Some people like to pull a loose tooth out in one yank, and others like to do ...

JPA/Hibernate

- 1) GET/Load
- 2) Update/Merge.
- 3) Criteria/ Reflections.
- 4) Joins/ Relations
- 5) Custom Generators.
- 6) What is the difference between CurdRepository and JPA repository? Which one you will prefer between the two

JWT

- 1) Json Web Token Structure .
- 2) Claims in Json Structure ..Explained custom claims
- 3) How are signatures verified in JWT ?
- 4) What form of Response do you get for JWT? Basically after encryption
- 5) Why are JWT headers necessary?

Hibernate

- 1) Difference between get and load?
- 2) Transaction management: how to handle it?
- 3) What is ddl ?
- 4) Session and session Factory?
- 5) Eager loading and lazy loading?

While lazy loading delays the initialization of a resource, eager loading initializes or loads a resource as soon as the code is executed. Eager loading ...

- 6) Open session and get current session?
- 7) What is a criteria builder?
- 8) Which is best 2nd level cache among these (testCache, EH Cache, GumoCache, DASCACHE)>>>Mostly EHcache is reffer
- 9) Can anyone please tell me how to fix the primary key violation constraint from code ?
- 10) What is the difference between 1st and 2nd level cache?
- 11) What will happen if we won't provide @id in entity class hibernate
- 12) Level 2 cache in hibernate (test, EH, Gummo,DAS) which is best

Hibernate second level cache uses a common cache for all the session object of a session factory.

The main difference between the first level and second level cache in Hibernate is that the first level is maintained at the Session level

- 13) In spring data jpa can we connect to two different DBs having two entity managers...is it possible if yes how ?
- 14) N+1 problem in hibernate.
- 15) Batch size in hibernate.
- 16) How to modify and save 1000 records in hibernate
- 17) User Hibernate's batch size property in application.properties file
- 18) What is the advantage of hibernate over jdbc
- 19) How will you save data to db using jdbc
- 20) How to remove duplicates from 10k records?
- 21) if one user is accessing some values from cache and the other user goes and makes manipulations in the database at the same time. It will lead to dirty reads. How to stop the first user from reading outdated data?
- 22) L1 and L2 Cache Difference?
- 23) What is @Query?
- 24) Is the user session and database session the same?
- 25) Difference between get() and load()?
- 26) Different Interfaces of Hibernate?
- 27) Difference between Save and persist?
- 28) Difference between Lazy and eager?
- 29) How to call stored procedures through hibernate?
- 30) How can we handle transactions in Jpa?
- 31) How can we invoke stored procedures in hibernate?
- 32) What's ordered collection and sorted collection
- 33) What is a persistent state?
- 34) How is dirty checking configured?
- 35) Configure callback interface?
- 36) How to make an object detached?
- 37) What happens if I detach a transient object?
- 38) explain each and every ORM level in Hibernate?
- 39) How to enable/disable first and second level cache?
- 40) How can we bind a hibernate session factory to JNDI ?
- 41) What's the difference between load() and get() method in hibernate?
- 42) difference between session.lock() and session.merge()?
- 43) How many concurrent Hibernate sessions can exist at any point in time in an application?
- 44) Can you explain the role Session interface plays in Hibernate?
- 45) How a class is mapped as immutable and why?
- 46) How can we use new custom interfaces to enhance functionality of built-in interfaces of hibernate?
- 47) if you don't want to save the state of the object what you will use?

Coding

rotate a 2d matrix by 90 degrees?

Wap to check whether the given substring is in the middle of the string or not?

Write a program on Singleton?

Which method will be called?

```
public show(int x){}
```

```
public show(short x){}
```

```

public show(long x){
main(){
    byte b= 1;
    show(b);
}

```

Create an api to get information corresponding to an ID, if present give result otherwise throw exception.

Sort array without using any predefined methods.

Solve using $O(n)$ or $O(n \log(n))$ Time complexity

Given nums=[2,7,11,15],target=9

Output =[0,1]

Explanation :: nums[0]+nums[1]

```

List<List<String>> l1 = Arrays.asList(Arrays.asList("pratik"), Arrays.asList("lalit"),
Arrays.asList("lokesh"), Arrays.asList("abhinav"));
Sort list l1 based on names

```

->Given five positive integers, find the minimum and maximum values that can be calculated by

summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

Example

arr = [1,3,5,7,9]

The minimum sum is 1+3+5+7=16 and the maximum sum is 3+5+7+9=24.

The function prints

16 24

->String str="hi hi hhii lkstjhihih hisss jjihi ji ji h i";

Find how many "hi" present in this string and possible tell me starting of the index

JUnit

1. What are you using for mocking the rest template? Annotations used?
2. How do you Mock DB?
3. Difference between junit 4 and 5?
4. Used JUnit and Mockito ?
5. What is the use of Mock Objects?
6. difference between mock and powermock
7. How do you mock void methods ?
8. How do you write test cases for handling exceptions ?

DevOps

1. What do you use for CI/CD? (Jenkin/GIT) explain?

React

1. What are your Roles and Responsibilities for the current project?
2. How comfortable are you with React ?
3. What are the advantages of Using React?
4. Tell me the complete flow of the application from UI to the DB.
5. How are you transferring data from react to services?
6. How and where are you deploying your project ?
7. Which service of IBM Cloud are you using for deployment?
8. Any experience with Angular, Node, HTML, CSS, JS, JQuery?
9. Async, promise, Observable, subscribers in react ?

Design Patterns

1. SOLID principle.
2. Singleton class how to implement?
3. Singleton
4. Template
5. Decorator
6. Prototype
7. Builder
8. MVC
9. Front Controller
10. View Resolver
11. DAO
12. Which design patterns have you used?
13. Diff blw singleton & prototype design patterns?
14. What is the Builder pattern?
- 15.

Data structures

problems with best case time complexity.

- 1) Returning distinct arrays from an array with duplicate entries.
- 2) Code for snake ladder game.
- 3) Find the largest possible palindrome from a given string.
- 4) Find the middle element of the linked list?
- 5) level order traversal of the tree?
- 6) What is merge sort and write program for it?

Sql

- 1) Find Employees by descending order of salary?
- 2) In which scenario we will use self join? Explain with an example?
- 3) Diff blw self join & full join in sql?
- 4) How to create stored procedures in DB?
- 5) Sql and NOsql which one is better?
- 6) What are the joins in SQL?
- 7) What is use of GROUP BY

