

2) What is Spring Boot?

→ Spring Boot is an open source, microservice-based Java web framework. The Spring Boot framework creates a fully production-ready environment that is completely configurable using its prebuilt code within its codebase.

3) MVC life-cycle in application ?

→ There are two life cycle in MVC

↳ Application life cycle

↳ Request life cycle

The starting point for every MVC application begins with routing. After that, the received request figures out and finds how it should be handled with the help of the URL Routing module.

4) Where do we write business logic and what is DAO?

→ @ service annotation write business logic.

business logic / business rules / data access should not be placed directly into models, views or controllers. To do so would be placing business logic in your presentation layer and thus reducing reuse and maintainability of your code.

DAO :- The Data Access Object (DAO)

Support in Spring is aimed at making it easy to work with data access technologies like JDBC, Hibernate, JPA or JDO in a consistent way.

Q) What is JPA Repository, and which dependency we need to add for data JPA?

→ JAVA Persistent API is the standard way of persisting Java objects into relational database.
four dependencies add in data JPA
spring web, spring data JPA; and H2 database

Q) What are features of Java 8?

- • Functional Interface and Lambda Expressions.
- forEach() method in Iterable Interface
- Default and static methods
- Optional class
- Java Date Time API
- Java Stream API for Bulk Data Operations on collections
- Collection API Improvements
- Java IO Improvements

Q) What is stream API in Java? what are diff methods used in it?

- - Stream API using new additional package java.util.stream package
- It is used to process collections of objects
- Basically stream is a sequence of objects that supports various methods
- * Diff methods : A stream is not a data structure instead it takes input from the collections, arrays or I/O channels provide the result as per the pipelined methods

8) `flatMap()` and `map()`
→ Stream interface has a `map()` and `flatMap()` methods and both have intermediate stream operation and return another stream as method output. Both of the functions `map()` and `flatMap` are used for transformation and mapping operations. `map()` function produces one output for one input value.

9) How `HashMap` works internally
→ Data is the most important part of an application, then data structure chosen to handle that data is even more important. Data structure arranges the data so that insertion of new elements or searching of old elements will be faster.

10) What technologies were used for front-end in your recent project?
→ HTML, CSS, JavaScript

11) Difference b/w abstract class & Interface

Interface	Abstract Class
1) It contains only abstract methods	1) It contains both abstract & non-abstract methods
2) It supports multiple inheritance	2) It does not support multiple inheritance
3) It gives 100% implementation	3) It gives partially (50%) implementation

12) Why to use interface over abstract class?

→ Abstract classes should be used primarily for objects that are closely related, whereas interfaces are best suited for providing common functionality to unrelated classes.

13) Default and static method in

→ - Default methods enable you to add new functionality to the interfaces of your libraries and ensure binary compatibility with code written for older versions of those interfaces.

- A static method is a method that is associated with the class in which it is defined rather than with any object.

14) What are functional interface? What is used for?

→ - functional Interface contain abstract method

- It can have default and static method

- functional Interface can have object class method

- It is in-built and predefined functional Interface

Used :- Functional interfaces are used and executed

by representing the interface with an annotation called `@FunctionalInterface`.

15) What are intermediate operations and terminal operations

→ The operations which return another stream as a result are called intermediate operations and the operations which return non-stream values like primitive or object or collection or return nothing are called terminal operations.

16) Try, catch and finally block?

- Try :- we specify the block of code that might give rise to the exception in a special block with a "try" keyword.
- catch :- when the exception is raised it needs to be caught by the program.
- finally :- sometimes we have an important code in our program that needs to be executed irrespective of whether or not the exception is thrown.

17) Checked and unchecked exception with example

- ① Checked Example :- checked exceptions are checked at compile time. It means if a method is throwing a checked exception then it should handle the exception using try catch block or it should declare the exception using throws keyword, otherwise the program will give a compilation error.
e.g :- FileNotFoundException

- ② Unchecked Example :- unchecked exceptions are not checked at compile time. It means if your program is throwing an unchecked exception and even if you didn't handle/declare that exception, the program won't give a compilation error.
e.g :- RuntimeException

18) How to customize own checked exception and own unchecked exception?

→ Checked exception :- To create a custom exception, we have to extend the java.lang.Exception class. We have to provide a constructor that takes a String as the error message and called the parent class constructor.

Unchecked exception :- we can create the custom unchecked exception by extending the RuntimeException in Java.

Unchecked exceptions inherit from the Error class or the RuntimeException class.

Q) Difference between HashMap and HashSet?



HashMap

1) HashMap is a hash table based implementation of map interface.

2) HashMap implements Map, Clonable, and Serializable interface.

3) It does not allow duplicate keys, but duplicate values are allowed.

HashSet

1) HashSet is a set. It creates a collection that uses a hash table for storage.

2) HashSet implements Set, Comparable, Serializable, Iterable and Collection interfaces.

3) It does not allow duplicate values.

20) Can we have duplicate keys in hashmap?

→ It does not allow duplicate keys, but duplicate values are allowed.

21) How Hashmap works internally?

- Hashmap belongs to map interfaces

- objects are stored in the form of key, value which is called Entry

- Entry is an internal class in Hashmap

- there are set of buckets, which store objects using linkedlist.

22) Difference between ArrayList & LinkedList

→

ArrayList

1) ArrayList internally uses a dynamic array to store the elements

2) ArrayList is better for storing and accessing data

3) An ArrayList class can act as a list only because it implements list only

LinkedList

1) LinkedList internally uses a doubly linked list to store the elements

2) LinkedList is better for manipulating data

3) LinkedList class can act as a list and queue both because it implements list and queue interfaces

23) What is authentication with examples?

→ Authentication is the process of verifying the identity of a person or device. A common example is entering a username and password when you log in to a website.

Entering the correct login information lets the website know 1) who you are and 2) that it is actually you accessing the website.

24) What is prototype bean and singleton bean?

→ prototype : A new instance will be created for single bean definition every time a request is made for that bean.

singleton : only one instance will be created for single bean definition per spring loc container and the same object will be shared for each request made for that bean.

25) Difference b/w "==" operator and .equals() method.

'==' operator	equals() method
1) == is considered an operator in java	1) Equals() is considered as a method in java
2) we can use the == op. with objects and primitives	2) we cannot use the equals method with primitives
3) == operator cannot be overridden	3) Equals() method can be overridden

26) Difference b/w Comparable and Comparator with example :-

Comparable	Comparator
↳ comparable provides natural sorting order	↳ comparator provides defn sorting order
↳ present in java.lang package	↳ present in java.util
↳ provide compareTo() method	↳ provide compare() method
↳ provide affects the original class	↳ does not affect the original class

27) Comparable eg

```
import java.util.*;
```

```
import java.io.*;
```

```
public Student implements Comparable<Student> {
```

```
int rollno;
```

```
String name;
```

```
int age;
```

```
Student (int rollno, String name, int age) {
```

```
this.rollno = rollno;
```

```
this.name = name;
```

```
this.age = age;
```

```
public int compareTo(Student st) {
```

```
    if (age == st.age)
```

```
        return 0;
```

```
    else if (age > st.age)
```

```
        return 1;
```

```
    else
```

```
        return -1;
```

```
}
```

```
}
```

→ public class Test {

```
public static void main (String args[]) {
```

```
ArrayList<Student> al = new ArrayList<Student>();
```

```
al.add (new Student(101, "Vijay", 23));
```

```
al.add (new Student(106, "Ajay", 27));
```

```
al.add (new Student(107, "Ravi", 21));
```

```
collection.sort (al);
```

```
for (Student st:al) {
```

```
System.out.println (st.rollno + " " + st.name + " " +
```

```
st.age);
```

```
}
```

```
}
```

```
}
```

O/P : 107 Ravi 21

101 Vijay 23

106 Ajay 27

27) What is classloader in Java?

→ 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.

three principles : Delegation, visibility, and uniqueness.

Types of classloader :- Bootstrap, Extensions,

28) Immutable class in Java
→ When we create an object of an immutable class, we cannot change its value.

29) Spring : How many different ways to create beans

→ 1. annotating your class with the stereotype

② component annotation

2. writing a bean factory method annotated with the
② Bean annotation in a custom Java configuration
class

3. declaring a bean definition in an XML configuration
file.

30) What is foreign key?

→ The foreign key constraint is used to prevent actions that would destroy links between tables.

81) different unix commands you use day to day life

→ ls command, cd command, cp command, mv command, rm command, mkdir command, rmrfie command, chown command.

82) what is prototype bean in spring

→ A new instance will be created for a single bean definition every time a request is made for that bean.

83) what is public and private keys in SSL hand shake process

→ All SSL certificates contain a key pair consisting of a public and private key.

The public key is used to encrypt data and the private key is used to decrypt.

84) what is equals and hashCode methods.

→ hashCode() : The hashCode() method returns the same hash value when called on two objects, which are equal according to the equals() method. And if the objects are unequal, it usually returns different hash values.

equals() : - It is used to compare two objects.

- By default, two objects will be the same only if stored in the same memory location.

35) Why ArrayList is better for read operations
→ because ArrayList provides constant time for search operation

36) Contract of Exception on method overriding
→ If when exception handling is involved with method overriding, ambiguity occurs. The compiler gets confused as to which definition is to be followed.

Types of problems :-

1. Problem 1 : If the superclass doesn't declare an exception
2. Problem 2 : If the superclass declares an exception

37) Write program to count number of elements in a string

```
public class Main {  
    public static void main (String args)
```

```
{  
    String str = "Hello world";
```

```
    int count = 0;
```

```
    System.out.println ("The entered string is : " + str);
```

```
    for (int i = 0; i < str.length(); i++)
```

```
{
```

```
    if (str.charAt(i) == ' ')
```

```
        count++;
```

```
{
```

```
System.out.println ("Total number of characters");
```

```
in the string : " + count);
```

3

3

o/p : The entered string is : Hello world
Total number of characters in the string : 10

38) Add element to given index in linkedlist.

→ import java.util.LinkedList;

```
public class InsertElement {  
    public static void main(String[] args) {
```

```
        LinkedList linkedList = new LinkedList();
```

```
        linkedList.add("element - 1");
```

```
        linkedList.add("element - 2");
```

```
        linkedList.add("element - 3");
```

```
        linkedList.add("element - 4");
```

```
        linkedList.add("element - 5");
```

System.out.println("Before inserting element - 6 at index 3, linkedList contains : " + linkedList);

```
        linkedList.add(3, "element - 6");
```

System.out.println("After inserting element - 6 at index 3, linkedList contains : " + linkedList);

3) Insert element after element at index 3.

3) Insert element at index 3.

Output - None (null pointer exception)

(last two are in bolded text in output)

39) What are different annotations in spring
→ `@Service`, `@Repository`, `@Component`, `@Autowired`,
`@Controller`, `@RequestMapping`, `@PostMapping`

40) Transaction levels in Hibernate
→ A transaction is an unit of work that helps to respect ACID principle.
1. read uncommitted
2. read committed
3. repeatable read
4. Serializable

41) Factory method
→ Define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate. also known as virtual constructor

42) String pool, final keyword?
→ • String pool :- special heap memory area
→ JVM will create an object in pool
• final keyword :- non-access modifier used for classes, attributes and methods, which makes them non-changeable
(impossible to inherit or override)