QUESTION BANK

1. OOPS concept (Inheritance, Polymorphism, Abstraction, Encapsulation) details.
2. Difference between .equals and ==.
3. Different ways to use a thread.

   In two ways the thred can be used

   A- By extending the thread class

   B- By implementing the Runnable Interface.
4. Java is Pass by value or Pass or reference.

   Java is always pass by value. During method invocation, a copy of each argument, whether its a value or reference, is created in stack memory which is then passed to the method.
5. Is there any other instance in Pass by reference.
6. Difference between Arraylist & Vector.

   ArrayList is non-synchronized where as the vector is synchronized.So arraylist performance is better than vector.
7. What is immutable class. Give example in Java and how we can create it.
8. . Difference between abstract class and interface.
9. Hash Map Vs Hash Set Vs Hashing.
10. Java 8 features.
11. Primitive Data Type Vs Object Data Type.
12. Final Vs Finally Vs Finalize.
13. If we don't override hashcode() method then what will happen.

    If you don't override hashcode() then **the default implementation in Object class will be used by collections**. This implementation gives different values for different objects, even if they are equal according to the equals() method.
14. Replace Vs Replace All vs ReplaceFirst.

    A-The **replace()** method is one of the most used string methods for replacing all the occurrences of a character with the given character. The replace() method of JDK 1.5 replaces the char and a sequence of char values.

    B- The **replaceAll()** method is similar to the String.replaceFirst() method. The only difference between them is that it replaces the sub-string with the given string for all the occurrences present in the string.

    C- The **replaceFirst()** method is another method for replacing the substring. It replaces the sub-string with the given string. The replaceFirst() method replaces only the first occurrence of the sub-string.

Types of Statements in java ?

Expression Statements

Expression is an essential building block of any Java program. Generally, it is used to generate a new value. Sometimes, we can also assign a value to a variable. In Java, expression is the combination of values, variables, operators, and method calls.

There are three types of expressions in Java:

- Expressions that **produce** a value. For example, **(6+9), (9%2), (pi*radius) + 2**. Note that the expression enclosed in the parentheses will be evaluate first, after that rest of the expression.
- Expressions that **assign** a value. For example, **number = 90, pi = 3.14.**
- Expression that **neither produces any result nor assigns a value**. For example, **increment** or **decrement** a value by using increment or decrement operator respectively, **method invocation**, etc.

Declaration Statements

In declaration statements, we declare variables and constants by specifying their data type and name. A variable holds a value that is going to use in the Java program. For example:

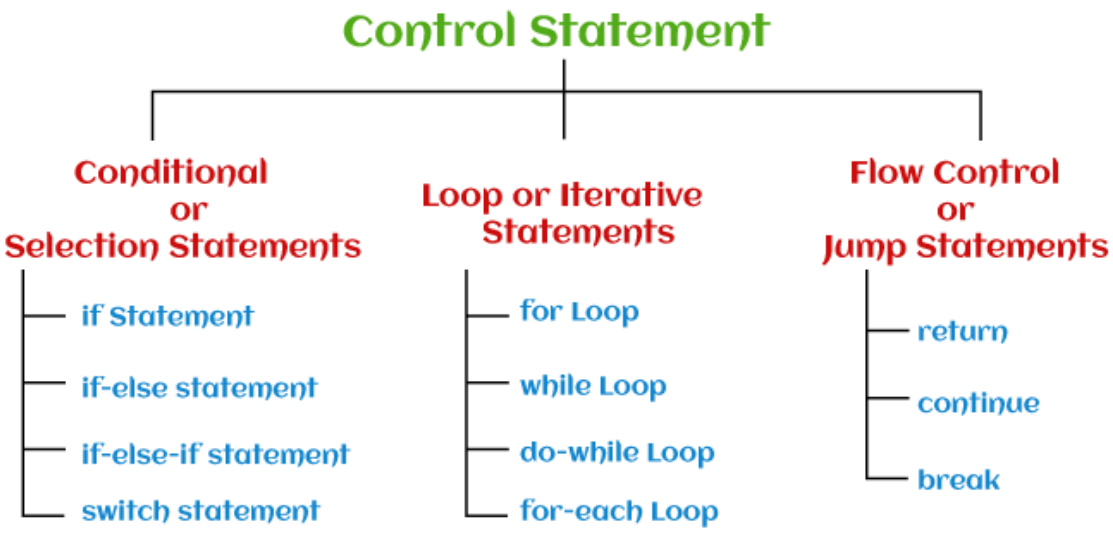
```
1. int quantity;
2. boolean flag;
3. String message;
```

Also, we can initialize a value to a variable. For example:

```
1. int quantity = 20;
2. boolean flag = false;
```

Control Statement

Control statements decide the flow (order or sequence of execution of statements) of a Java program. In Java, statements are parsed from top to bottom. Therefore, using the control flow statements can interrupt a particular section of a program based on a certain condition.



PROGRAMS->

A. Different logical questions on String (.equals and ==).

B. Occurrence of character in a given string.

C. Find length of the given string.

D. Sorting of Array.

E. Palindrome of string, integer.

F. Armstrong Number

G. Sort an ArrayList.

H. In String find how many time occurrence of characters.

I. Remove duplicate characters from String.

SQL->

- Foreign Key Vs Primary Key.
- Drop Vs Delete Vs Truncate Command.
- Joins and their types.
- DDL Vs DML.

DDL	DML
It stands for Data Definition Language.	It stands for Data Manipulation Language.
It is used to create database schema and can be used to define some constraints as well.	It is used to add, retrieve or update the data.
Basic command present in DDL are CREATE, DROP, RENAME, ALTER etc.	BASIC command present in DML are UPDATE, INSERT, MERGE etc.
DDL does not use WHERE clause in its statement.	While DML uses WHERE clause in its statement.

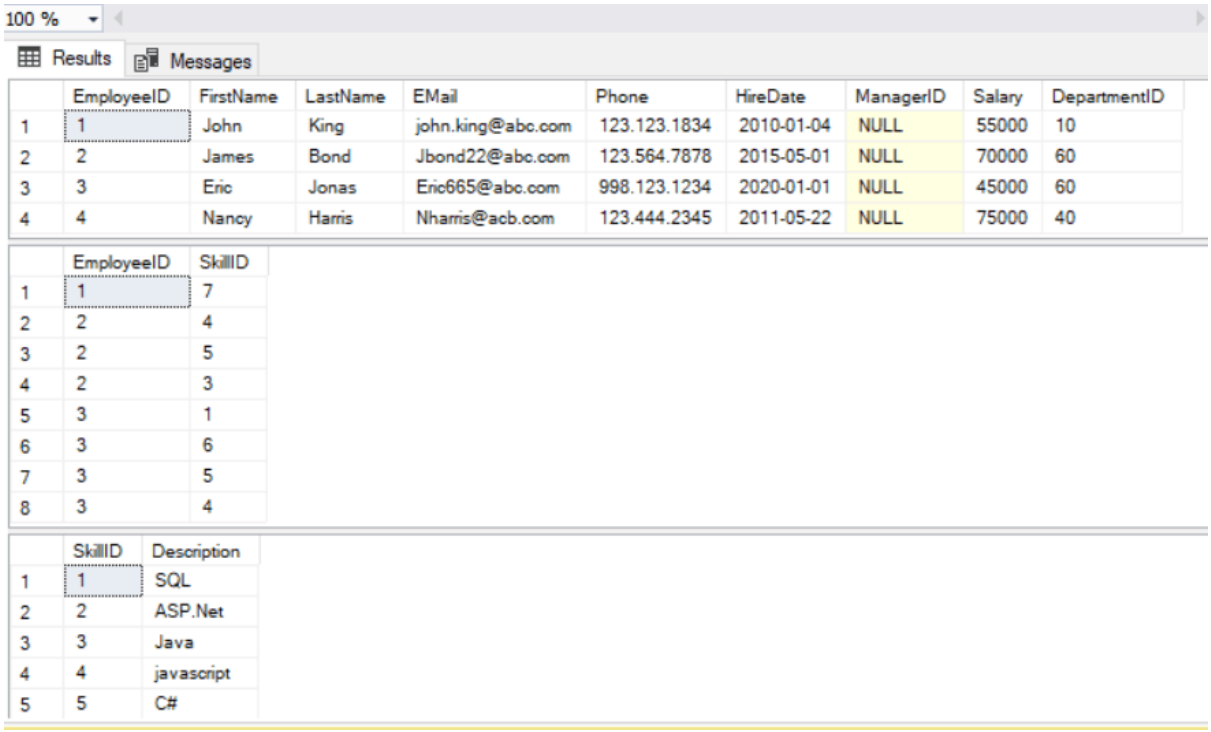
- DBMS Vs RDBMS
- SQL Vs MYSQL Vs NOSQL
- Normalizations

  Normalization is the process to eliminate data redundancy and enhance data integrity in the table. Normalization also helps to organize the data in the database. It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables.
- Relationship in SQL and its types.

  In One-to-One relationship, one record of the first table will be linked to zero or one record of another table.

  In One-To-Many relationship ,A single record from one table can be linked to zero or more rows in another table

  In Many-To-Many relationship, there is a junction table created which holds the primary keys of another tables and marked as foreign keys.



100 %

Results | Messages

	EmployeeID	FirstName	LastName	EMail	Phone	HireDate	ManagerID	Salary	DepartmentID
1	1	John	King	john.king@abc.com	123.123.1834	2010-01-04	NULL	55000	10
2	2	James	Bond	Jbond22@abc.com	123.564.7878	2015-05-01	NULL	70000	60
3	3	Eric	Jonas	Eric665@abc.com	998.123.1234	2020-01-01	NULL	45000	60
4	4	Nancy	Harris	Nharris@acb.com	123.444.2345	2011-05-22	NULL	75000	40

	EmployeeID	SkillID
1	1	7
2	2	4
3	2	5
4	2	3
5	3	1
6	3	6
7	3	5
8	3	4

	SkillID	Description
1	1	SQL
2	2	ASP.Net
3	3	Java
4	4	javascript
5	5	C#

- How to fetch common records from two tables.
- How to select unique records from a table.
- What is index in SQL.

  Indexes are used to retrieve data from the database very fast. The users cannot see the indexes, they are just used to speed up searches/queries.

  The `CREATE INDEX` command is used to create indexes in tables (allows duplicate values).

  Query- `CREATE INDEX idx_lastname ON Persons (LastName);`
- 5.What is View in SQL.

  In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are filled from one or more real tables in the database.

  Query-

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

- What s Cursor in SQL.

  Cursor is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of Performing DML(Data Manipulation Language) operations on Table by User.

There are 2 types of Cursors: Implicit Cursors, and Explicit Cursors. These are explained as following below.

1. **Implicit Cursors:**
   Implicit Cursors are also known as Default Cursors of SQL SERVER. These Cursors are allocated by SQL SERVER when the user performs DML operations.
2. **Explicit Cursors :**
   Explicit Cursors are Created by Users whenever the user requires them. Explicit Cursors are used for Fetching data from Table in Row-By-Row Manner.

- CRUD operation quaries.
- How remove duplicates in SQL.

  Query-

```
WITH CTE AS (
SELECT Name, Email, City
ROW_NUMBER() OVER (
PARTITION BY Name, Email. City
ORDER BY Name, Email. City
) row_num
FROM Geek
)
DELETE FROM CTE
WHERE row_num > 1;
```

17. SPRING BOOT->

A. Spring Vs Spring Boot (Advantages, Disadvantages, Which one is better).

B. Spring Initializer.

The Spring Initializr is ultimately a web application that can generate a Spring Boot project structure so that we can import that into the IDEs like eclipse…

C. Spring Acuator.

Spring Boot Actuator is **a sub-project of the Spring Boot Framework**. It uses HTTP endpoints to expose operational information about any running application.

The main benefit of using this library is that we get health and monitoring metrics from production-ready applications.

D. Spring Boot starter.

Spring Boot Starters are **dependency descriptors that can be added under the <dependencies> section in pom. xml**. There are around 50+ Spring Boot Starters for different Spring and related technologies. These starters give all the dependencies under a single name.

E. Spring MVC.

A Spring MVC is a Java framework which is used to build web applications. It follows the Model-View-Controller design pattern. It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

F. Flow of Spring MVC.

G. Advantages of Spring MVC over other framework.

1. Excellent support for developing RESTful web services.
2. **Separate roles** - The Spring MVC separates each role, where the model object, controller,view resolver, DispatcherServlet, validator, etc. can be fulfilled by a specialized object.
3. **Light-weight** - It uses light-weight servlet container to develop and deploy your application.
4. **Rapid development** - The Spring MVC facilitates fast and parallel development.
5. **Flexible Mapping** - It provides the specific annotations that easily redirect the page.

H. Declaration of different layer in Spring Boot.

I. How to connect Spring Boot application to database using JDBC.

k. What is spring starter parent dependency tag.

L. 6.How you exposing data and how do you consuming it.

M. Beanfactory vs Applicationcontext .

One difference between the bean factory and application context is that the former only instantiates bean when you call getBean() method while ApplicationContext instantiates Singleton bean when the container is started, It doesn't wait for the getBean to be called.

N.Types of Bean Scope.

1. SingleTone
2. prototype
3. request
4. session
5. Global Session

P.Types of Auto wiring.

1. no
2. ByName
3. byType
4. Constructor
5. AutoDetect

18. HIBERNATE->

A. What is ORM ?

B. Hibernate Vs JDBC.

JDBC	Hibernate
It is database connectivity technology	It is a framework,
It does not support lazy loading	Hibernate support lazy loading
We need to maintain explicitly database connection and transaction.	Hibernate itself manage all transaction
Low performance	High Performance

C. Session and SessionFactory in hibernate.

SessionFactory is a class for Session objects. It is available for the whole application while a Session is only available for particular transaction.

Session is short-lived while SessionFactory objects are long-lived.

SessionFactory provides a second level cache and Session provides a first level cache.

D. Lazy Loading in hibernate.

Lazy loading is a fetching technique used for all the entities in Hibernate. It decides whether to load a child class object while loading the parent class object.

E. Caching in hibernate .

Caching is a mechanism to enhance the performance of a system. It is a buffer memory that lies between the application and the database. Cache memory stores recently used data items in order to reduce the number of database hits as much as possible.

F. How to create immutable class in hibernate.

1. **Avoid providing any methods which modify object state.** Obvious candidates are property setters as well as any other methods adjusting existing properties.
2. **Make all fields private** – to avoid modifying them directly, especially if they are reference variables.
3. **Make all fields final** – to explicitly express intent that their values should not change. This also means all the properties need to be assigned at the moment of creation in constructor.
4. **Ensure class cannot be extended** – this eliminates a possibility to expose its variables indirectly through a child class. It can be accomplished by making the class final or by providing a private constructor. Static factory method or a builder class is used to instantiate objects in such a scenario.

G. Hibernate Architecture.

The Hibernate architecture is categorized in four layers.

- Java application layer
- Hibernate framework layer
- Backhand api layer
- Database layer

H. IOC Container.

The IoC container is responsible to instantiate, configure and assemble the objects.

There are two types of IoC containers. They are:

1. **BeanFactory**
2. **ApplicationContext**

I. get() Vs load().

Get()	Load()
It is used to fetch data from the database for the given identifier	It is also used to fetch data from the database for the given identifier
If object not found for the given identifier then it will return null object	It will throw object not found exception
It returns fully initialized object so this method eager load the object	It always returns proxy object so this method is lazy load the object
It is slower than load() because it return fully initialized object which impact the performance of the application	It is slightly faster.
If you are not sure that object exist then use get() method	If you are sure that object exist then use load() method

J. Hibernate vs JDBC. Why Hibernate over JDBC

K. All Hibernate annotations.

M. Lazy Loding vs Eager Loading.

Lazy	Eager
In Lazy loading, associated data loads only when we explicitly call getter or size method.	In Eager loading, data loading happens at the time of their parent is fetched
Many-To-Many and One-To-Many associations used lazy loading strategy by default.	Many-To-One and One-To-One associations used eager loading strategy by default.
It can be enabled by using the annotation parameter : fetch = FetchType.LAZY	It can be enabled by using the annotation parameter : fetch = FetchType.EAGER
Initial load time much smaller than Eager loading	Loading too much unnecessary data might impact performance

O. States of Object in Hibernate.

1.Transient State

An object we haven't attached to any *session* **is in the transient state.** Since it was never persisted, it doesn't have any representation in the database. Because no *session* is aware of it, it won't be saved automatically.

2.Persistamce State-

An object that we've associated with a ***session*** is in the persistent state. **We either saved it or read it from a persistence context, so it represents some row in the database.**

3.Detached State-

When we close the *session*, all objects inside it become detached. Although they still represent rows in the database, they're no longer managed by any *session*.

Q. First Level Cache Vs Second Level Cache.

20. OTHER QUESTIONS ->

Is it possible to change default port number.

B. Microservices And Webservices .

C. How to integrate email API in project.

D. Which all API'S you used in your project.

E. Why would you prefer STS on other applications.

21. (Project explanation).

What is static, non-static/instance, local variables?

22. Difference between primitive and non-primitive data types?
23. What is Constructors and its types and need of constructor?
24. What is SOLID principle with advantages?
25. Difference between JDK, JRE and JVM?
26. What is anonymous class?

An **anonymous inner class** is an inner class which is declared **without any class name** at all. In other words, a nameless inner class is called an **anonymous inner class**. Since it does not have a name, it cannot have a **constructor** because we know that a constructor name is the same as the class name.

Types of Anonymous Inner Class in Java

- Anonymous inner class that extends a class
- Anonymous inner class that implements an interface
- Anonymous inner class as an argument

27. What is Garbage collector and need of garbage collector in java?
28. Explain Types of memory in java and explain each?
29. What are Class, Method and Object in java?
30. Difference between return and return value?
31. What is this keyword?
32. What are packages in java?

packages as being similar to different folders on your computer.

Package in Java is **a mechanism to encapsulate a group of classes, sub packages and interfaces**.

Packages are used for: Preventing naming conflicts. For example there can be two classes with name Employee in two packages, college. staff.

33. What is type casting and its type explain each?
34. What is interface and abstract class and when to use interface and when to use abstract class with example?
35. What is abstract keyword?
36. What is run time polymorphism?
37. What are marker interface and its uses and need?

Marker interface is **used as a tag that inform the Java compiler by a message so that it can add some special behavior to the class implementing it**.

38. What are java 8 features explain each?

39. What are access modifier explain each?
40. Can we change the visibility of method while overriding
41. If yes then what is condition in it?
42. What are Scanner class and its uses?
43. What is for loop?
44. Difference between While and Switch statement?
45. Difference between continue and break statement?
46. Explain main method in java?
47. What is System.out.println()?Explain
48. What is Array? And its type
49. What is Serialization with example?
50. What is Deserialization with example?
51. What is transient keyword?
52. What is volatile keyword?
53. What is var type?
54. What is string tokenizer?

StringTokenizer	Split()
It is a legacy class that allows an application to break a string into tokens.	It is a method of the String class or the java.util.regex package that splits this string around matches of the given regular expression.
It returns one substring at a time.	It returns an array of substrings.
It can't handle empty strings well.	It can handle empty strings when you need to parse empty tokens like *ant, bat, pat*

55. Can we overload static method in java?
56. Can we override static method in java?
57. What is super most class in java?
58. What is data hiding and how to achieve?
59. What is Static block in java?

    It is a block in java which runs only once when a class is sloaded into JVM,In other word it is also called as Static Intitialization block.
60. What is Non-Static block in java?
61. What is Super keyword in java?
62. Difference between this and Super keyword?

63. What are Exception in java and its type explain each?
64. How many ways to handle exception in java explain each?
65. Explain exception hierarchy?
66. Can we write finally without catch block?
67. Difference between throws and throw?
68. What is exception propagation?
69. Explain Checked and unchecked exception?
70. What are command line arguments?
71. What is thread?

Threads allows **a program to operate more efficiently by doing multiple things at the same time**. Threads can be used to perform complicated tasks in the background without interrupting the main program.

72. What is Demon thread?

A Daemon thread is **a background service thread which runs as a low priority thread and performs background operations like garbage collection**.

73. What is multithreads?

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.

Threads can be created by using two mechanisms :

1. Extending the Thread class
2. Implementing the Runnable Interface

74. Difference between thread join and sleep method?

There is a difference between join() and sleep(). **join() will wait until the timeout expires or the thread finishes. sleep() will just wait for the specified amount of time unless interrupted**.

75. Process to create thread and which is best and why?
76. What is thread scheduler?

Thread scheduler in Java is **the component of JVM that determines the execution order of multiple threads on a single processor (CPU)**. It decides the order in which threads should run. This process is called thread scheduling in Java.

77. Explain about priority of thread?

Each thread has a priority. Priorities are represented by a number between 1 and 10. In most cases, the thread scheduler schedules the threads according to their priority (known as pre-emptive scheduling).

Note that not only JVM a Java programmer can also assign the priorities of a thread explicitly in a Java program by using setPriority() method..

78. Explain wait, notify and notify all?

The threads can communicate with each other through **wait(), notify()** and **notifyAll()** methods in Java.

These are **final** methods defined in the **Object** class and can be called only from within a **synchronized** context.

The **wait()** method causes the current thread to wait until another thread invokes the **notify()** or **notifyAll()** methods for that object. The **notify()** method **wakes up a single thread** that is waiting on that object's monitor. The **notifyAll()** method **wakes up all threads** that are waiting on that object's monitor.

79. What is thread pool?

80. What are executor services?

The Java ExecutorService is the interface which allows us to execute tasks on threads asynchronously. The Java ExecutorService interface is present in the java.util.concurrent package.

The ExecutorService helps in maintaining a pool of threads and assigns them tasks.

It also provides the facility to queue up tasks until there is a free thread available .

81. Difference between runnable and callable interface?

Runnable interface	Callable interface
It is a part of *java.lang* package since Java 1.0	It is a part of the *java.util.concurrent* package since Java 1.5.
It cannot return the return of computation.	It can return the result of the parallel processing of a task.
It cannot throw a checked Exception.	It can throw a checked Exception.
In a runnable interface, one needs to override the run() method in Java.	In order to use Callable, you need to override the call()

82. Advantage and Disadvantage of thread?

The `Thread` class has the following advantages:

- Threads can be utilized to free up the main thread.
- Threads can be used to break up a task into smaller units that can be executed concurrently.

The `Thread` class has the following disadvantages:

- With more threads, the code becomes difficult to debug and maintain.
- Thread creation puts a load on the system memory and CPU .

- We need to do exception handling inside the worker method as any unhandled exceptions can result in the program crashing.

83. Explain thread life cycle?

New: Whenever a new thread is created, it is always in the new state. For a thread in the new state, the code has not been run yet and thus has not begun its execution.

Active: When a thread invokes the start() method, it moves from the new state to the active state. The active state contains two states within it: one is **runnable**, and the other is **running**.

- **Runnable:** A thread, that is ready to run is then moved to the runnable state. In the runnable state, It is the duty of the thread scheduler to provide the thread time to run, i.e., moving the thread the running state.

- **Running:** When the thread gets the CPU, it moves from the runnable to the running state. Generally, the most common change in the state of a thread is from runnable to running and again back to runnable.

Blocked or Waiting: Whenever a thread is inactive for a span of time (not permanently) then, either the thread is in the blocked state or is in the waiting state.

Terminated: A thread reaches the termination state because of the following reasons:

- When a thread has finished its job, then it exists or terminates normally.
- **Abnormal termination:** It occurs when some unusual events such as an unhandled exception or segmentation fault.

84. What is Synchronization?

Synchronization is the process to get control over multiple threads.

85. What is Synchronized keyword?

86. What is String in java and why String is Immutable in java?

87. Difference between String, stringbuffer and stringbuilder?

88. What do you mean by mutable and immutable?

89. How to make a class immutable with example?

90. Why java is platform independent language?

91. Why java is not pure object oriented language?

92. Why pointers are not used in java?

A pointer is just the address of some memory location.

Java doesn't support pointer explicitly, But java uses pointer implicitly: Java use pointers for manipulations of references but these pointers are not available for outside use.

Any operations implicitly done by the language are actually NOT visible

93. What is singleton class and how we can make a class singleton?

94. What is Collection framework?

A collection framework is one which is used for storing and manipulating a group of objects.

95. What are wrapper classes?

96. What is cloning?

97. What is hash code and equal method?

98. What is JDBC and write code of JDBC?

99. Difference between Collection and Collections?

Collection:

Collection is a interface present in java. util.package. It is used to represent a group of individual objects as a single unit.

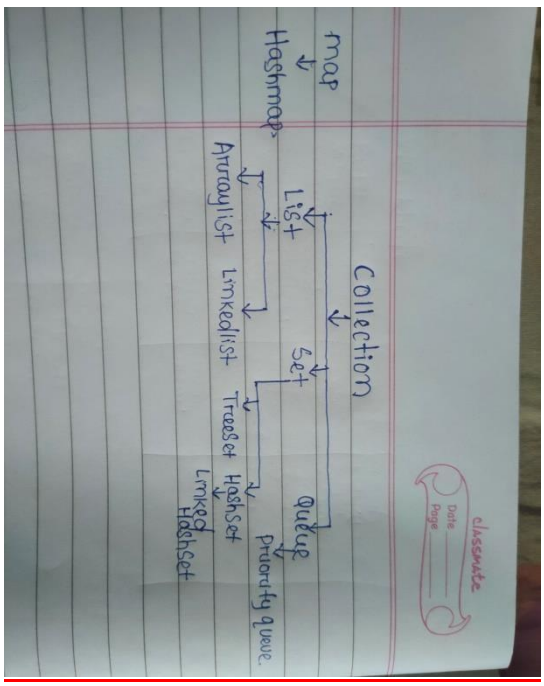
The List, Set, and Queue are the main sub-interfaces of the collection interface.

The **add()**, **remove()**, **clear()**, **size()**, and **contains()** are the important methods of the Collection interface.

Collections:

Collections is a utility class present in java.util.package. It defines several utility static methods like sorting and searching which is used to operate on collection.

101.) Explain Collection hierarchy? Explain List, Set and Queue?



1) List: - list is an interface. List follows insertion order. It consist of duplicate elements. To use list interface we can use three classes. **i. Arraylist**

The `ArrayList` class is a resizable array, which can be found in the `java.util` package.it is internally work like a dynamic array.

ii. Linkedlist

We can use linkedList if there is no continuous memory of location. Adding data is easier than arraylist.it divided into 2 types – i)single LinkedList , ii)Double LinkedList.

iii. vector

It work same as array but it is synchronized and give lower performance than ArrayList

2) **Set** : Set interface doesn't follow any insertion order and can't contain duplicate element. We can use set interface by 2 classes. i.Hashset

It uses hashtable internally.it contains only unique element. It is synchronized and it doesn't permit null element. It is the parent class of LinkedHashSet.

LinkedHashSet use for remove duplicate element from an array.

ii.Treeset

It is use for sort data in ascending order.

3)Queue:-

Queue is the parent interface of priority queue. Priority queue is follows FIFO(first in first out) order for adding and removing data. **102. Difference between ArrayList vs LinkedList and Vector?**

ArrayList stores in continuous memory location where LinkedList doesn't required a continuous memory location. ArrayList is non-synchronized where vector is Synchronized

103. Explain HashSet and HashMap?.

HashMap	HashSet
Data stores as key-value pair.	Data stores as object.
synchronized	Non-synchronized
Doesn't allow duplicate element	It contains single null key and multiple null values.
It is slower than hashSet	Faster than HashMap.
Implement of set interface	Implement of map interface.

104. Internal working of HashMap?

It internally uses hashtable. It uses hashcode of the key object and this hashcode used to find index of the bucket.

105. What is Collision and how to overcome this?

It is a situation where two or more key objects produce the same final hash value. To overcome it each bucket consist of

linkedlist in it.

106. Difference between synchronized hashmap and concurrent hashmap?

Synchronized hashmap	concurrent hashmap
synchronized but slower than concurrent hashmap	synchronized but faster than synchronized hashmap
It locks whole map	It locks some portion of map

107. Difference between comparable and comparator?

Comparable	comparator
1) Comparable provides a **single sorting sequence**. In other words, we can sort the	The Comparator provides **multiple sorting sequences**. In other words, we
collection on the basis of a single element such as id, name, and price.	can sort the collection on the basis of multiple elements such as id, name, and price etc.

108. Difference between Enumeration and Iterator?

Enumeration	Iterator
Only consist read()	Have both methods remove(),read()
Access only vector and hashtable	Access any class in collection.

109. What is Ternary operator with example?

The ternary operator is an operator that exists in some programming languages, which takes three operands rather than the typical one or two that most operators use.

Ex:- < ,> ,? ,==

110. What are unary operators and its example?

which takes one operands i.e. single input it is use to increment or decrement value.

ex:- ++, --

111. What are association and its type?

An **association** defines a relationship between two entity objects based on common attributes. **The** relationship can be one-to-one or one-to-many.

112. Difference between Iterator and List Iterator?

List iterator is child interface of iterator.

Iterator travel in one direction having method next(), hasNext(). List Iterator travel in both direction having method previous(), hasPrevious(),next(), hasNext().

113. What is Fail fast and fail safe?

Fail-fast and Fail-safe are the iterators or collections in Java.

The major difference is fail-safe iterator doesn't throw any Exception like fail-fast.

114. What is Servlet?

Servlet is a java class.it helps us to perform backend coding it interact with frontend view and backend business logic.

115. Difference between DoGet and DoPost?

DOGET	DOPOST
Getting data from database	Submit data into database

Data expose to URL	Data not expose to URL
When you refresh page you'll not get any security pop-up alert	When you refresh page you get security pop-up alert

116. What is InterServlet Communication?

Here one servlet can call from another servlet.by using getRequestDispatcher

117. What is Request Dispatcher.

It receive request from client and send it to any resources like:- servlets, jsp or HTML file.

118. Difference between ServletConfig and ServletContext?

ServletConfig	ServletContext
It represent single servlet	It represent whole web application

119. Explain Servlet Life cycle?

For the 1st time when we start tomcat init method will runs once.(usually used to connect DB)

DoPost and DoGet can run several times.

120. What are Session Variables and its type?

It is a special type of variable which value can be access by across the web pages.

There are four mode types or just modes. In-Process mode, State Server mode, SQL Server mode, Custom mode and Off mode.

121. What is JSP and its tag?

JSP stands for java server page.

We can write partially java code in it by using jsp tags. There are 4 types of jsp tags

1.**scriptlet tag** :- (<% %>)

It uses implicit object like:-session,request,response,out. Can't use HTML code inside it and can't declare any variable or method inside it.

2.**Declaration tag** :-(<%! %>)

Doesn't have any implicit object. Can declare variable and method.

Can use access specifiers.

3.**Expression tag :- (<= >)**

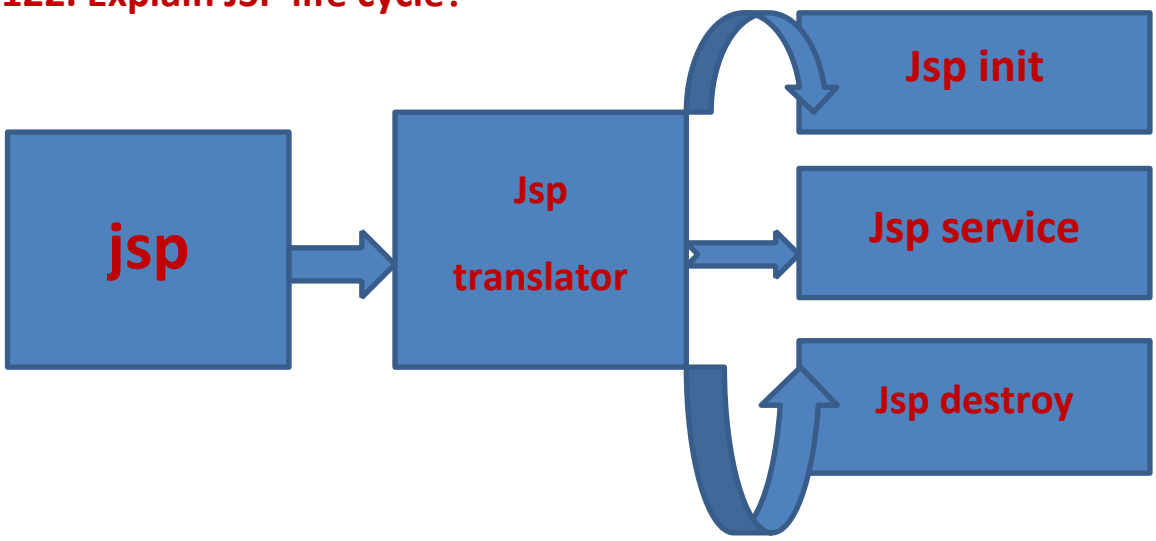
Can write only one statement of code. used for print output

4.**Directive tag :(<@ >)**

It is divided into 2 types :- page and include

Page directive is use to import any packages into the jsp page and include directive use to import any file into the page

122. Explain JSP life cycle?



Jsp init can invoked once .jsp service() is used to service the raised request by jsp .it takes request and response object as parameters.

123. What is JSTL tag?

Jstl stands for jsp server tag library .we can write java code in it, in form of html tag.

124. What are implicit object?

These are the readily available object present in jsp page ex:-session ,response ,request ,out .etc.

125. What is HTML and feature of HTML5?

Html stands for hypertext markup language . It is a pre-defined markup language use to create User Interface(view page).HTML 5 is the new version where audio and video are supported.

126. Explain MVC architecture?

MVC architecture pattern separates an application into three main logical component model ,view and controller.

View is use Html ,jsp ,jstl to make user Interface. Model is used to perform business logic in it.

Controller layer is used to interact between view and model.

127. Difference between spring and SpringBoot?

spring	SpringBoot
Complex to use	Easy to use
For hibernate you've to download all dependency	Only have to download starter tag
Configuration done in xml file	Configuration done in application. Properties file

128. What is Web Services and its type?

It helps us to integrate heterogeneous and homogenous application .

Generally it is 2 types:-i) S0AP (simple object access protocol) , ii)

Rest.

129. How to expose and consume web services?

If you want to exchange data between two application by json or xml file you have to expose

data to xml/json then another application will consume this data.

130. Difference between SOAP and RESTFull web services?

SOAP	RESTFull
Data exchange using xml file	Data exchange using both xml file & json object
Complex to implement through xml file	Simple to use json object

131. What is POSTMAN?

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs

132. What are Http methods explain each?

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP methods are:- I)GET :- fetch data from DB and display it as json object.

II)POST :- Save data to the DB using web services. III)PUT :- update record to DB using web services.

IV)Delete :-Delete record in DB using web services

133. Difference between micro services and monolithic applications?

In monolithic application all module based on one server. In micro services bigger application break into smaller mini project and interact by web services

134. What is Maven?

Maven is a build tool which is based on the concept of a project object model (POM),. Using maven, we can build and manage any Java-based project.

135. What is Spring Initializer?

Spring initializer gives spring boot project structure which we can further use to make spring boot project in eclipse.

136. What is Unit testing?

It is also known as White Box testing where testing done by code level (programming level).

Ex.:- jUnit ,testing.

137. What is Junit?

Junit used for unit testing. Annotation used in junit are:-

@Test,@Before,@After,@BeforeTest,@AfterTest

138. What is JPA?

Java persistence API(JPA) is a concept of taking the object content and map it into the Database by ORM (object relationship mapping).

139. What is Hibernate JPA?

The implementation of JPA can be done by using Hibernate.

140. What are Spring IOC and Its type?

Inside Spring IOC(inversion of control) dependency injection logic written inside it. There are 2 types of Ioc available I)BeanFactory ,

II)ApplicationContext

141. What is spring security?

It is a framework that focuses on providing authentication and authorization mechanism to spring application.

142. What are JPA annotations used in your project explain each ?

JPA annotation are:-

(i)@Entity :-It defines which java class is map to DB table. **(ii)@Table :-**When DB table and Entity class name isn't same we use this annotation for mapping.

(iii)@Column:- This is used to map entity class variable into DB column

(iv)@Id :- It maps entity class variable with primary key column of DB.

(v)@Generated Value:- It helps us to auto-increment value of variable in entity class while saving record in DB.

143. What are spring boot annotation used in your project explain each?

Spring boot annotation are:-

@Autowired,@controller,@service,@component,@RequestMappin,
@RequestParam,@RestController,@GetMapping,@PostMapping,
,@PutMapping,
,@DeleteMapping,@pathvariable,@modelAtteribute,@RequestBody
,@Qualifier,@springBootApplication

144. What is Dependency injection?

It is core of spring framework. Used to inject bean inside
Given reference variable

145. Types of Autowired and explain each?

I) Default

II)By name

III)By type

IV) constructor

146. Difference between @Controller and @RestController?

@controller define controller layer in spring boot , @Restcontroller defines webservices layer in our project.

147. Difference between @Autowiring and @Bean?

When interface implement only one class @Autowiring used When interface implement more than one class @Bean used.

148. What is @Qualifier and @Services and @Component and @Value and @Query ?

@Qualifier :- if one interface implement into 2 different classes ,so we've to tell spring which class object should create.

@Service :- Define the service layer of spring boot.

@component :- It tells spring boot to maintain object life cycle.

@Query :-It use to declare query directly on repository method.

149. Explain @SpringBootApplication?

It defines the starting point of execution in spring boot project.

150. Difference between BeanFactory and ApplicationContext?

The applicationContext comes with advanced features,while beanFactory comes with basic features.

Application context implement from bean factory interface.

151. Difference between JPA and Hibernate?

JPA is a concept where we taking object content and map it into the

DB and implementation done by hibernate.

152. Explain advantage of Hibernate over JDBC?

Jdbc is a connectivity ,hibernate is a framework

Jdbc has lower performance than hibernate We can create database

table by using hibernate

153. What is Session in hibernate?

Session interface(API) is the main tool used to communicate with hibernate It used to get a

physical connection with Database.

154. What are roll back and commit?

Once rollback is execute ,the database would reach its previous state.

Once the commit statement execute ,the data cannot be rolled back

155. Session management in spring boot?

I) create spring boot project from spring initializer. ii)add spring session jdbc

dependency in pom.xml iii)add spring jdbc properties in application. Properties.

156. How to create session object?

HttpSession mySession=request.getSession( );

157. Difference between session and sessionFactory?

Session factory is a factory class for session objects.it available for whole application while

session only available for particular transaction.

158. Difference between get and load method of hibernate?

Both are used to fetch data .

Get( ) return null if no row is available on the database

Load( ) throws object not found exception.

159. What is caching and hibernate and explain 1st level and 2nd level caching?

Caching is the mechanism to enhance threperformance of the system.

1st level maintained at session level and accessible only to the session, while 2nd level maintain at the session factory level and available to all sessions.

160. What are Core interface of hibernate?

- Configuration interface
- Transaction interface
- Session interface - SessionFactory interface **161. What is Jenkins?**

Jenkins is used for testing and reporting.

162. What is JIRA?

Jira is a software application used for issue tracking and project management.

163. How to read application. Properties in any class?

Read a property from application. Properties file using @ Configuration Properties & @value

164. What are response codes of POSTMAN?

200/201/204/400/401/403/404/405/500/503

200= ok

201=file created

400= bad request

404=file not found

500=internal server error

503=service unavailable

203=Non- authorized

501=HTTP server error

165. Explain Session Tracking System

Session Tracking is a way to maintain state (data) of an user.

166. Explain types of spring bean scope?

Singleton , prototype ,request ,session, global session

167. What is Actuator in spring boot?

Actuator is used to expose operational information .It is the subproject of spring boot.

168. What is Starter tag in spring boot?

It is the default configuration of all hibernate dependency.

169. Types of hibernate mapping explain each?

Primitive type mapping (integer ,long , short ,Boolean)

Date and time mapping (Date , time ,timestamp ,calendar

Binary and Large Object Types(binary ,text)

Jdk related type (class,timezone)

170. Types of inheritance in hibernate?

Hibernate supports 3 types of Inheritance Mappings: Table per class hierarchy

Table per sub-class hierarchy

Table per concrete class hierarchy

171. Difference between Drop and Truncate?

Drop :- It is used to eliminate the whole database from the table.

Truncate :- It is used to eliminate the data from the table

172. Find 2nd max salary from employee table?

To find 2nd max salary we use sub-query.

Select max(salary) from employee where salary<(Select max(salary) from employee where salary)

173. What is Primary key and foreign key?

Primary key can have unique value but not null.

Foreign key helps us to build relation between 2 tables it consists of repeated value

174. Explain SDLC model?

There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "**Software**

Development Process Models

175. Explain about Agile methodology?

We can build a software at aleast possible time without compromise quality.

We generally use scrum process for that **176. Explain JDK, JRE and JVM?**

Java development kit (jdk) consist of development kit with jre. It is used by developer to write and run programme.

Java runtime environment (jre) consist 0f java library and a jvm in it

.It can used by customer to run the programme.

Jvm is read the .class file and give the output.

177. Explain public static void main(String args[]) in Java

Public :- jvm call the .class file from another package

Static :- main method can call without creating an object.

Void :- main method has no return type,

String args[] :- it is command line argument.

178. Why Java is platform independent?

When .java file compiled into .class file it convert into byte code by JIT compiler

.and this byte code can use any platform because of jvm.

179. Why Java is not 100% Object-oriented?

Because without creating an object we can access the data by through static block

180. What are wrapper classes in Java?

It stores the primitive data type value inside an object, by doing this we get many in-built method for manipulating the value.

181. What are constructors in Java?

Constructors are used to initialize the object we created. These are permanently internally void

.

182. What is singleton class in Java and how can we make a class singleton?

In these class only one object can create throughout the class.

To design a singleton class: 1 Make constructor as private. 2 Write a static method that has return type object of this singleton class.

183. What is the difference between Array list and vector in Java?

Working of array list and vector are same but vector is synchronized where array list is non-synchronized.

184. What are the differences between Heap and Stack Memory in Java?

In stack memory execution of programme done and in heap memory object will created. Stack is faster than heap because it follows LIFO

(last in first order)

185. What is Polymorphism?

Polymorphism is a oops concept .here we develop such a feature that a method can have many form. We can achieve polymorphism by 2 ways .:-Overriding, Overloading.

186. What is runtime polymorphism or dynamic method dispatch?

Class upcasting done with method overriding known as rutime polymorphism.

187. What is abstraction in Java?

In abstraction we hide internal implementation details and showing only functionality to the user. We can achieve it by interface and abstract class.

188. What do you mean by an interface in Java?

Interface consist of only incomplete method in it. We can use it by implement it into a class . Multiple inheritance supported in interface but we can't create object inside in an interface.

189. What is the difference between abstract classes and interfaces?

Interface consist of only incomplete method where abstract class consist both complete and incomplete method .multiple inheritance done by using interface.

190. What is inheritance in Java?

In inheritance we inherit the properties of an object from parent class to child class . Inheritance done for reusability.

191. What are the different types of inheritance in Java?

(i)Single inheritance :- one parent class → one child class

(ii)Multiple inheritance :- multiple parent class-→one child class (iii)Multilevel inheritance :- 1^{st} parent class→2^{nd} parent class/1^{st} child class→2^{nd} child class

(iv)Hybrid class :- multiple +multilevel inheritance.

192. What is method overloading and method overriding?

In method overriding we inherit a method and change its logic in child class . In method overloading we can create more than one method in same class with different no of argument and different types of arguments.

193. Can you override a private or static method in Java?

NO ,we can't inherit private or static method in java

194. What is multiple inheritance? Is it supported by Java?

When more than one classes have only one child class called as multiple inheritance. In class level multiple inheritance doesn't support but in interface level it supports.

195. What is encapsulation in Java?

When we binding all the data and code together that is known as encapsulation. We use getter and setter for perform encapsulation.

196. What do you mean by aggregation?

In the case of Aggregation, an object can exist without being part of the main object.

197. What is composition in Java?

A composition in Java between two objects associated with each other exists when there is a strong relationship between one class and another.

198. What is a marker interface?

A empty body interface called marker interface. Ex -Seralizable ,clonable ,etc.

199. What is a constructor overloading in Java

If more than one constructor is there inside one class with different no. of arguments or different type of argument

Is known as constructor overloading.

200. What is a servlet?

Servlet is a class used to perform backend coding. It used for interact between view and model

Servlets – Java Interview Questions

What is Request Dispatcher?

100. What is the life-cycle of a servlet?

101. How does cookies work in Servlets?

102. What are the different methods of session management in servlets?

JDBC – Java Interview Questions

103. What are the steps to connect to a database in java?

104. What is the difference between execute, executeQuery, executeUpdate?

Spring - Java Interview Questions

105. Explain Bean in Spring and List the different Scopes of Spring bean.

106. Write a program for a thread safe singleton class?

107. What is autowiring in Spring? What are the autowiring modes?

108. How to handle exceptions in Spring MVC Framework?(@ExceptionHandler).

109. Which one will you prefer among spring and springboot to create a program from scratch?

Hibernate-Interview Questions

110. What is Hibernate Framework?

Ans:- hibernate is an open-source object-relational mapping (ORM) based java persistence framework. It is an ORM mapping tool in java. Hibernate is designed with the need to reduce complexity while connecting a relational database through java. Hibernate framework is designed to map java objects to implement object-oriented programming in the relational database.

a. What are the advantages of Hibernate over JDBC ?

Ans:- **Advantages of Hibernate over JDBC:**

1. Hibernate is an ORM tool
2. Hibernate is an open source framework.
3. Better than JBDC.
4. Hibernate supports inheritance and polymorphism.
5. With hibernate we can manage the data stored across multiple tables, by applying relations(association)
6. Hibernate has its own query language called Hibernate Query Language. With this HQL hibernate became database independent.
7. Hibernate supports relationships like One-To-One, One-To-Many, Many-To-One ,Many-To-Many.
8. Hibernate supports lot of databases.
9. Hibernate supported databases List.
10. Hibernate also supports annotations along with XML.
11. Hibernate supports Lazy loading.
12. Hibernate is easy to learn it is developers friendly.
13. Hibernate maintains database connection pool.
14. Using Hibernate its Easy to maintain and it will increases productivity

111. What is JPA can you directly implement it?

JSP Interview Questions

112. JSP lifecycle?

113. What are the JSP implicit objects?

114. What are the different tags in JSTL?

Ans:- The JSTL tags can be classified, according to their functions, into the following JSTL tag library groups that can be used when creating a JSP page −

- **Core Tags**
- **Formatting tags**
- **SQL tags**
- **XML tags**
- **JSTL Functions**

115. Explain the jspDestroy() method.

Exception and Thread - Interview Questions

116. What is the difference between Error and Exception?

117. How can you handle Java exceptions?

118. What are the differences between Checked Exception and Unchecked Exception?

119. Checked Exception?

120. Define Threads and its types?

121. Will the finally block get executed when the return statement is written at the end of try block and catch block as shown below?

122. Thread life cycle?

123. What are the differences between throw and throws?

124. How to create a custom Exception?

125. What is a finally block? Is there a case when finally will not execute?\

126. Can we write multiple catch blocks under single try block?

127. What is OutOfMemoryError in Java?

Ans:- OutOfMemoryError is a subclass of java.lang.VirtualMachineError in java. It is thrown by the Java Virtual Machine (JVM) when an object cannot be allocated due to lack of memory space and also, the garbage collector cannot free some space.

128. What is a Thread?

129. What are the two ways to create a thread?

130. Garbage collectors in Java?

131. Can you call a constructor of a class inside another constructor?

132. How is the creation of a String using new() different from that of a literal?

133. Why is synchronization necessary? Explain with the help of a relevant example.

134. What are the differences between Heap and Stack Memory in Java?

- Ans:- The major difference between Stack memory and heap memory is that the stack is used to store the order of method execution and local variables while the heap memory stores the objects and it uses dynamic memory allocation and deallocation.

135. What is JIT compiler in Java?

- Ans:- The Just-In-Time (JIT) compiler is a component of the JRE (Java Runtime Environment) that improves the performance of Java applications at run time. It helps improve the performance of Java programs by compiling bytecodes into native machine code at runtime .

136. What are access modifiers in Java?

- Ans:- The access modifiers in java specifies accessibility (scope) of a data member, method, constructor or class. There are 4 types of java access modifiers: private. default. protected. public.

137. What is an object in Java and how is it created?

Ans:- a. An entity that has state and behavior is known as an object .

b. There are five different ways to create an object in Java:

- Java new Operator
- Java Class.newInstance() method
- Java newInstance() method of constructor
- Java Object.clone() method
- Java Object Serialization and Deserialization

138. What is Object Oriented Programming?

- Ans:- The object-oriented programming is basically a computer programming design philosophy or methodology that organizes/ models software design around data, or objects rather than functions and logic. An object is referred to as a data field that has unique attributes and behavior. Everything in OOP is grouped as self-sustainable objects.

139. What are the main concepts of OOPs in Java?

140. What is the difference between a local variable and an instance variable?

141. What is an infinite loop in Java? Explain with an example.

Ans:- Infinite loop in java refers to a situation where a condition is setup so that your loop continues infinitely without a stop. A loop statement is used to iterate statements or expressions for a definite number of times but sometimes we may need to iterate not for a fixed number but infinitely.

Ex- int i=o;

While(i<1000) {

i*1; }

142. What is Java String Pool?

143. Differentiate between static and non-static methods in Java.

144. What is constructor chaining in Java?

145. Difference between String, StringBuilder, and StringBuffer.

146. What is a classloader in Java & Types?

Ans:- A ClassLoader is an object responsible for dynamically loading Java class during runtime to prevent JVM from realizing that ClassLoader is a part of the Java Runtime Environment. It makes JVM life easier. JVM loads the classes into memory when required by the application and does not load all at once. ClassLoader then comes into the picture and loads the class into memory.

A Java Classloader is of **three types**:

1. **BootStrap ClassLoader**
2. **Extension ClassLoader**
3. **System ClassLoader**

147. Why Java Strings are immutable in nature?

Ans:- In Java, string objects are immutable in nature which simply means once the String object is created its state cannot be modified. Whenever you try to update the value of that object instead of updating the values of that particular object, Java creates a new string object.

148. What is the difference between an array and an array list?

Ans:- The **main difference** between Array and ArrayList is that **Array is a fixed length data structure while ArrayList is a variable length Collection class.**

149. collection hierarchy below go throught all of them i have attached the image below.

150. Internal Algorithm working of Hashmap?

- Ans:- Internal Working of HashMap in Java. The function 'hashCode' is used to get the hash code of an object in Java. This is an object of super class Object. It returns the object reference's memory as an integer. It is a native function, which means no direct method in Java can be used to fetch the reference of the object.

151. Define webservice and microservices briefly?

152. What methods were you using in REST?

Ans:- Following four HTTP methods are commonly used in REST based architecture.

- **GET** − Provides a read only access to a resource.
- **POST** − Used to create a new resource.
- **DELETE** − Used to remove a resource.
- **PUT** − Used to update a existing resource or create a new resource.

153. Methods of Hibernate?

Ans:- Methods of Hibernate Session

1. **Save():** Save() method generates the primary key and inserts the record in the database. It is similar to the persist() method in JPA but it behaves differently in a detached instance by creating the duplicate record upon database commit.
2. **Update():**Update() is used to update the existing database record. It returns an exception if the record is not found or called in a transient instance.
3. **saveOrUpdate():** It saves or updates the database based on the entity passed. It does not return an exception in the transient state but it makes the state to persistent during a database operation.
4. **merge():** Values from a detached entity are updated to the database when the merge() is used by changing the detached entity to the persistent state.
5. **delete():** Delete method works in persistent mode to remove the entity from the database. An exception is returned if no record is found in the database.

154. Xml v/s JSON?

155. Miscelleneous Question

156. Classnotfound Exception v/s Noclassdef found Exception?

Ans:- Both *ClassNotFoundException* and *NoClassDefFoundError* occur when the JVM can not find a requested class on the classpath. Although they look familiar, there are some core differences between these two.

ClassNotFoundException

ClassNotFoundException is a checked exception which occurs when an application tries to load a class through its fully-qualified name and can not find its definition on the classpath.

This occurs mainly when trying to load classes using *Class.forName()*, *ClassLoader.loadClass()* or *ClassLoader.findSystemClass()*.

NoClassDefFoundError

NoClassDefFoundError is a fatal error. It occurs when JVM can not find the definition of the class while trying to:

- Instantiate a class by using the *new* keyword
- Load a class with a method call

The error occurs when a compiler could successfully compile the class, but Java runtime could not locate the class file.

157. Intermediate and Terminal operations in stream?

Ans:- 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.

158. Group by & order by?

159. Delete v/s Drop?

160. SIB v/s IIB?

161. Steps to create immutable class in java?

162. Map v/s Flat Map?

Ans:- flatMap would perform mapping each array not with stream but with the contents of that stream. All of the individual streams that would get generated while using map(Arrays::stream) get merged into a single stream.

163. Reduce() method?

Ans:- In Java, **reduce()** is a method of the **Stream** interface.

It is the method of combining all elements. For each element presented in the stream, the reduce() method applies the binary operator. In that stream, the first argument to the operator must return the value of the previous application and the second argument must return the current stream element.

It allows us to produce a single result from a sequence of elements by repeatedly applying a combining operation to the elements in the sequence called **reducing**.

164. What is spring boot starter?

Ans:- **Spring Boot** provides a number of **starters** that allow us to add jars in the classpath. Spring Boot built-in **starters** make development easier and rapid. **Spring Boot Starters** are the **dependency descriptors**.

In the Spring Boot Framework, all the starters follow a similar naming pattern: **spring-boot-starter-***, where * denotes a particular type of application. For example, if we want to use Spring and JPA for database access, we need to include the **spring-boot-starter-data-jpa** dependency in our **pom.xml** file of the project.

Spring Boot Actuator Features

There are **three** main features of Spring Boot Actuator:

- **Endpoints**
- **Metrics**
- **Audit**

165. Status codes of postman?

Ans:- A status code defines the status of the request. On entering URL, a mistake can be typed in the URL, or there may be a server-side problem. Status code is used to know about what went wrong and where you made a mistake. There are different status codes, and each of the status codes has different meanings.

166. Different errors codes like 404,400,500,200?

Ans:- **404 Not Found** : The requested resource was not found. Most commonly found error.

400 Bad Request : The request was incorrectly done.

500 Internal Server Error : This is a collection-status code for unexpected server error.

An HTTP status code 200 means success.

167. Lazy loading v/s Eager loading?

168. Iterators and types?

Ans:- An `Iterator` is an object that can be used to loop through collections, like ArrayList and HashSet. It is called an "iterator" because "iterating" is the technical term for looping.

- Types:-
- ***Enumeration*** *− Enumeration is initial iterators introduced in jdk 1.0 and is only for older collections like vector or hashTables. Enumeration can be used for forward navigation only. Element can not be removed using Enumeration.*
- ***Iterator*** *− Iterator is a universal iterator introduced in Jdk 1.2 and can be used for any collections. Iterator can b e used for forward navigation only. Element can be removed using iterator if remove method is supported.*
- ***ListIterator*** *− ListIterator is a iterator for List type collections and supports bidirectional navigation.*

169. List iterator?

Ans:- It is used to read, remove, add, and update the collection elements by iterating over specified List type collection.

170. Session, session factory?

Ans:- **SessionFactory** is an interface. SessionFactory can be created by providing Configuration object, which will contain all DB related property details pulled from either hibernate.cfg.xml file or hibernate.properties file. SessionFactory is a factory for Session objects.

We can create one SessionFactory implementation per database in any application. If your application is referring to multiple databases, then you need to create one SessionFactory per database.

The SessionFactory is a heavyweight object; it is usually created during application start up and kept for later use. The SessionFactory is a thread safe object and used by all the threads of an application.

A Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database. Persistent objects are saved and retrieved through a Session object.

The session objects should not be kept open for a long time because they are not usually thread safe and they should be created and destroyed them as needed. The main function of the Session is to offer, create, read, and delete operations for instances of mapped entity classes.

171. Spring MVC?

Ans:- A Spring MVC is a Java framework which is used to build web applications. It follows the Model-View-Controller design pattern. It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

A Spring MVC provides an elegant solution to use MVC in spring framework by the help of **DispatcherServlet**. Here, **DispatcherServlet** is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.

172. JPA anotations?

Ans:- @Entity, @Table ,@Column ,@Id ,@GeneratedValue.

173. Stream API with example?

Ans:- The Java Stream API provides a functional approach to processing collections of objects. The Java Stream API was added in Java 8 along with several other functional programming features.

Ex- We can find the even numbers,square numbers from a number list.

Like - public class A{

Psvm(){

```
List<Integer> data =Arrays.asList(1,2,3,4);

List<Integer> evenNum=data.Stream().filter(x->x%2==0).collect(Collectors toList());

Sop(evenNum);
```

174. Microservices v/s Monolithic architecture?

175. what is DDL and DML?

Ans:- **The key distinction is that the DDL command is used to create a database schema, while the DML command is used to modify the table's existing data**.

DDL Commands- DDL stands for Data Definition Language.

The following commands come under DDL language:

- **CREATE**
- **DROP**
- **ALTER**
- **TRUNCATE**: It is used to completely remove all data from a table, including their structure and space allocates on the server.
- **RENAME**: This command renames the content in the database.

DML Commands- It stands for Data Manipulation Language.

The following commands come under DML language:

- **SELECT**: This command is used to extract information from a table.
- **INSERT**: It is a SQL query that allows us to add data into a table's row.

- **UPDATE**: This command is used to alter or modify the contents of a table.
- **DELETE**: This command is used to delete records from a database table, either individually or in groups.