Hi,

 Please brush-up below points before the further discussion.

* Sort a map by value,
* Create immutable class,
* Diff between abstract class and interface
* Print numbers from 1 to 20 using two threads
* Java 8 features - optional, stream
* Executor framework - implementation and usage
* Multithreading and concurrency
* Java Collections
* Design Principles and Patterns
* Serialization, Reflection
* Joins, Simple Queries, Normalization,
* Immutability
* Synchronization
* Inheritance, Polymorphism

1. Exception hierarchy in Java. Should clear confusion on when to use which type of Exception in Java. Why catching Throwable instead of hierarchy wise exception classes is a bad idea.

2. Focus on Generics. Concepts of Type Erasure, specifying generic types for classes and type restrictions.

3. Knowledge about basics of Java 8. How does Java 8 changes the role of interfaces. Why are abstract classes still relevant in Java 8.

4. Core Java mixed with questions that would require good working knowledge on Algorithms.

5. Checking if a very large number is prime or not

6. Questions based on Singly and doubly linked list. Comparison with Array list and questions on doing binary search on Linked list.

7. Questions on Exception handling in Java. Questions on when do we need a checked vs unchecked exception in Java. Questions on when to extend Runtime Exception vs Exception for custom exceptions. Questions on Errors and Throwable.

8. Questions on Interface and Abstract Methods on Java. Questions on how Java 8 changes interfaces.

9. Questions on Generics. Concepts of Type erasure, making classes that can accept generic types. Questions on putting type restrictions for classes.

10. Questions based on Java collection framework. Lists vs Sets, their performances in concurrent environments and collection choices for implementing algorithms based on searching, sorting, slicing and traversing through data.

**Java collections, data structures and theory complexities including linear/hashing/tree, multithreading/Concurrency, Databases, SQL and stored procedures**

Datastructures - linked list, stack, queue, heap data structure

Database and SQL query

**Problem solving questions :-**

1. Given 2 lists of integers A, B. Find A-B
2. 2.Sort a stack. Again was able to solve it even after putting some constraints
3. 2.Sort a stack. Again was able to solve it even after putting some constraints

<http://javahungry.blogspot.com/2013/08/hashing-how-hash-map-works-in-java-or.html>

<http://javahungry.blogspot.com/2014/06/how-treemap-works-ten-treemap-java-interview-questions.html>

<http://javahungry.blogspot.com/2015/02/how-concurrenthashmap-works-in-java-internal-implementation.html>

<http://javahungry.blogspot.com/2015/10/how-treeset-works-internally-in-java-interview-questions.html>

<http://javahungry.blogspot.com/p/threads.html>

<http://javarevisited.blogspot.in/2011/04/synchronization-in-java-synchronized.html>

<http://javarevisited.blogspot.in/2011/06/volatile-keyword-java-example-tutorial.html>

<http://mrbool.com/working-with-java-executor-framework-in-multithreaded-application/27560>

<http://stackoverflow.com/questions/10828863/what-the-use-of-custom-class-loader>

<http://stackoverflow.com/questions/10901752/what-is-the-significance-of-load-factor-in-hashmap>

<http://stackoverflow.com/questions/11011291/treeset-internally-uses-treemap-so-is-it-required-to-implement-hashcode-method>

<http://stackoverflow.com/questions/137975/what-is-so-bad-about-singletons>

<http://stackoverflow.com/questions/13855013/understanding-java-memory-management>

<http://stackoverflow.com/questions/2087469/sort-a-file-with-huge-volume-of-data-given-memory-constraint>

<http://stackoverflow.com/questions/27325997/how-does-countdownlatch-works-in-java>

<http://stackoverflow.com/questions/56860/what-is-the-liskov-substitution-principle>

<http://stackoverflow.com/questions/8161896/example-code-to-show-how-java-synchronized-block-works>

<http://tutorials.jenkov.com/java-concurrency/synchronized.html>

<http://tutorials.jenkov.com/java-util-concurrent/cyclicbarrier.html>

<http://www.dynatrace.com/en/javabook/how-garbage-collection-works.html>

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/G1GettingStarted/index.html>

<http://www.programcreek.com/2013/03/hashmap-vs-treemap-vs-hashtable-vs-linkedhashmap/>

<https://docs.oracle.com/cd/E13150_01/jrockit_jvm/jrockit/geninfo/diagnos/garbage_collect.html>

<https://en.m.wikipedia.org/wiki/Creational_pattern>

<https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)>

<https://lostechies.com/derickbailey/2009/02/11/solid-development-principles-in-motivational-pictures/>

<https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>

**Java Basics**   
  
1 "What is stack and heap in java?   
Jvm parameter tuning?   
if your java goes into the full gc?" Worked on java profiling and performance testing.   
2 What is REST architecture? What is stateless api?   
3 "What is dependency injection?  
How do we achieve it in java?"

**Multithreading and problem solving**

1 Create your own connection pool? Couldnt identify datastructure   
2 "What is a Reentrant lock?  
Why it is called reentrant?  
Can the same thread hold lock twice?"

3 What is AtomicInteger? Do you know the internal implementation? Dont know internal mplementation   
  
**SQL**   
1 ways-to-find-nth-highest-salary   
  
  
**Data structures**1 Prefix search. Sample Ar gives Arc, ArchiTecture, Art, Artist, Artisan Using trie.   
2 Check if array is formed of consecutive elements? Gave O(n)

**Angular and Javascript questions :-**

1. Change detection - on push and default
2. View child & ElementRef
3. Dynamic components and Component lifecycle hooks
4. Routing
5. State management - Service, Subjects and Behavior Subjects
6. Design - Whatsapp example to create modules, components, routes for that
7. Javascript and ES6 knowledge

* JS - Scoping, Closures, Functions, Prototypes
* JS - Event loop, callbacks, promises, async/await   
  JS - Advanced features, other tools   
  NG - Typescript
* NG - Components   
  NG - Routing, Animation
* NG - Event Handling, RxJS, State
* Immutability
* Memory model and GC
* Synchronization
* sleep vs wait
* Orderids in time range

1. Why were ORMs like Spring data JPA , Hibernate introduced over JDBC?
2. If you have to set up JDBC in your application, what all changes would you do in your application.
3. How do you manage transactions within your application?
4. Suppose I am calling 4 different services from my application. From 1st service I am fetching data, 2nd service I am updating data into Database 3rd is doing some delete operation followed by 4th service update operation. Now all these operations are to be managed within single transaction. How will you do it. Assume these services have their own databases. Can it be done by using spring transaction management. If yes how, if now then what is the alternative.
5. (i) You have 2 synchronized methods within a class and 2 threads. Can both the threads access the 2 methods at the same time? In what scenario the threads can access the 2 methods together. In what scenario the threads cannot access the 2 methods together and only one can be accessed?

(ii) Now I want more than 1 thread holding lock on same instance to access the same synchronized block. What will you do for that?

(iii) I want 2 threads holding lock on same instance to access the 2 methods together using locks. How can it be done?

1. Perform Serialization and Deserialization of binary tree. Discuss what all scenarios can come and how you will manage serialization and deserialization in all these scenarios for the binary tree.
2. (i)What do you understand by Singleton?

(ii) Why is it necessary to make certain objects singleton in nature?

(iii)What kind of objects have you made singleton in your application?

(iv) If I would have used Static class with the restriction within the project that no one is allowed to create more than 1 object for this class ,would it serve my purpose? Can I replace static class in place of Singleton in such a scenario.

1. (i)What are the prerequisites of making a custom class object as HashMap key . What all changes you would do in your custom class to make such a scenario possible. What is the issue caused if we don’t override equals method?

(ii) Suppose I have not overridden hashcode and equals methods in my custom class. Will it work properly?

(iii) Will the get method provide me the correct and desired data from map if both equals and hashcode are not overriden? If yes , what is making the get method work properly? If no , then what will be the output

1. Is SimpleDateFormat a threadsafe object ? If not then how do you make sure it is threadsafe in nature? What is the exception you get when multiple threads access the SimpleDateFormat in a concurrent way ? What is the exception you get when multiple threads access a collection like arraylist ?
2. Can a static method access class level variables(instance variables) directly?
3. (i)Write a query to find the employees having same salary in an Employee table.

(ii) Is there a difference in performance of query if you do Select S1.name from Student s1,Student s2 where …. instead of Select S1.name from Student s1 join Student s2 on s1.id = s2.id where….

1. How do you define multiple beans within your application?
2. Suppose you are having a huge number of beans within your application. How would you maintain such a large number of beans in your application .
3. How would you maintain general POJOS within your application ?
4. Suppose there are 2 component classes which implement the same interface . Now you want to autowire both ? Will it cause a runtime exception ?If yes , How can you resolve such an issue? If no , how spring manages such a scenario?
5. Is map a part of collection hierarchy?
6. How is Garbage collection performed in java? What makes an object survice GC and move from eden space to survivor spaces at the time of minor GC event?
7. (i) What all changes you can think of in order to make a custom class object as hashmap key? Why immutability is considered as a preferred option when it come to hashmap key?

(ii) Suppose I have an Employee class with id and some other field. I have overridden the equals and hashcode methods. Within equals method implementation I have not used id and have used remaining fields only . Will it cause me any problem? In what scenarios it will work and it what scenarios it wont work.

(iii) Now if I remove both equals and hashcode methods from Employee class will it work for me. In what scenarios it will work and it what scenarios it will fail ?

(iv)Suppose I want to use a 3rd party api object which doesn’t override equals and hashcode as a key in hashmap. How can I do it?

(v)The api is having a class Employee with a method of getDetails(Employee emp) and doesn’t have equals and hashcode overriden. I want my application to access that method . How can I access that method .

1. (i) What kind of threading mechanisms you have implemented in your application?

(ii) How do you decide which threadpool to use for what scenario within your application?

(iii)How do you decide what is the number of threads that threadpool should be having?

(iv) Suppose there are 100 tasks within your application what should be the the ideal size of threadpool in such a case.

(v)Suppose there are many tasks which keep on coming to your application with no particular DB operations nor external 3rd party service interactions . What kind of threadpool you would be using in such a scenario. What should be ideal size to your threadpool?

1. **Coding ques** : A genius guy has predicted the cost of 1 type of shares for 30 days in 1 go. Some days the cost is going up some days it is going down. Assume that he can buy at most 10 shares on a given day and sell any no of shares he is holding. Design a algorithm in such a way that at by the end of 30 days he should be earning maximum profit .
2. You have been given a list of items . Find out the average price of these items using java 8 code.

Q1 IMPLEMENTATION CODING

double findHalfAreaPoint(double X)

e.g. if areaUnderCurve(X) returns 100, then findHalfAreaPoint(X) should return a value "H" such that areaUnderCurve(H) will return 50.

NOTE: areaUnderCurve is an API function which you can call.

\*/

/\* Q2 SQL

Table: Hobbies

        Person | Hobby

        ----------------

        John         | Movies

        John         | Baseball

        Sameer       | Reading

        Sameer       | Music

        Sameer       | Movies

        Write a query for:

        1. Names of all the people who have the hobby of reading. // Parth

        2. Names of all the people who do not have the hobby of reading.\*/

/\*Q3

\* 3.1 MAP AND CONCURRENT HASHMAP

\* 3.2 CONCURENT HASHMAP AND TREEHASHMAP

\* 3.3 THREAD SAFE BEHAVIOUR OF ABOVE DS

\* 3.4 BUCKET LOCKING CONCEPT

Q4

About current project

JUNIT TESTING

CHALLANGES OF MOCKING UNIT CASES

Java-

* Network Overriding
* Memory Consumption
* Memory Allocation

From his projects-

* How thread has worked in that particular project
* Scenario of Multithreading in that project

Design a system based on hierarchy of the employees structure.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1) Two classes having static method with same name and if one class extends another.

For eg: Class B extends Class A and suppose both class have method name : display().

What will happen and will it override in base class?

2) Relationship betweem hashcode and equals method during hashing techniques.

3) User defined class has been created, lets say Student {age,name}and muliple instances have been created s1,s2,s3,s4 etc.

This instances have been stored in a HashMap key and what if we modify the contents of one key lets say s2. Where will be the key stored? Explain.

4) There is a class with 2 static synchronized methods M1 and M2 and Multiple threads are accessing these methods. How will they access these methods? Explain.

5) There is a List of integer values and need to use some methodologies to get even integers from list and its sum using JAVA 8 techniques.

6) Define a body of function with input parameter as a paragraph String. Write a logic to count the number of words and count of each words if it occurs more than once.

7) There is a list of values indicating per day stock price. For eg: 1-100,2-150,3-200.......30-100 [day-stockPrice]. Now write a logic to find a day when customer should purchase a stock and a day when he should sell to get a maximum profit. return buy-day,sell-day and Profit Price.

8) DB Query considering 3 tables Student[id , name], Subject[id, name], Marks[student\_id,Subject\_id, Marks].

Query 1: Return [Student Name, Subject Name, Marks]

Query 2: Return [Student Name,Total Marks]

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1) Given a function with input parameter as a List<String>.            Write a logic to remove duplicates from that List.

2) Given a function with input parameters as (List<Integer>, List<Integer>). Return the elements which are distinct from both the lists.

3) Given a User defined class Employee {String name, float sal} and have overriden hascode method which returns same value for all inputs. What will happen of instances of Employee class is put into a HashMap as a key.

Scenarios 2: if Employee class overrides equals method and also it returns true for all values. What will happen of instances of Employee class is put into a HashMap as a key.

4) Design a Web browser such that,

a) User can open tabs such  as Tab1, Tab2, Tab3 etc.

b) If user follows a pattern to open Tab1-> Tab2-> Tab3 and goes back to open Tab1 and then closes Tab3 then it should directly opens Tab1 as it is accessed before Tab3.

c) Design a algorithim and which data structure will be used to solve the problem in an efficient way.

5) Given a Function with input parameter as 2D-Array of Integers with any order. A logic is written inside a function to put these values into Database table.

Design a Table schema to handle dynamic change in order of input matrix.

Please find the below list of Java concepts and links

* Sort a map by value,
* Create immutable class,
* Diff between abstract class and interface
* Print numbers from 1 to 20 using two threads
* Java 8 features - optional, stream
* Executor framework - implementation and usage

<http://javahungry.blogspot.com/2013/08/hashing-how-hash-map-works-in-java-or.html>

<http://javahungry.blogspot.com/2014/06/how-treemap-works-ten-treemap-java-interview-questions.html>

<http://javahungry.blogspot.com/2015/02/how-concurrenthashmap-works-in-java-internal-implementation.html>

<http://javahungry.blogspot.com/2015/10/how-treeset-works-internally-in-java-interview-questions.html>

<http://javahungry.blogspot.com/p/threads.html>

<http://javarevisited.blogspot.in/2011/04/synchronization-in-java-synchronized.html>

<http://javarevisited.blogspot.in/2011/06/volatile-keyword-java-example-tutorial.html>

<http://mrbool.com/working-with-java-executor-framework-in-multithreaded-application/27560>

<http://stackoverflow.com/questions/10828863/what-the-use-of-custom-class-loader>

<http://stackoverflow.com/questions/10901752/what-is-the-significance-of-load-factor-in-hashmap>

<http://stackoverflow.com/questions/11011291/treeset-internally-uses-treemap-so-is-it-required-to-implement-hashcode-method>

<http://stackoverflow.com/questions/137975/what-is-so-bad-about-singletons>

<http://stackoverflow.com/questions/13855013/understanding-java-memory-management>

<http://stackoverflow.com/questions/2087469/sort-a-file-with-huge-volume-of-data-given-memory-constraint>

<http://stackoverflow.com/questions/27325997/how-does-countdownlatch-works-in-java>

<http://stackoverflow.com/questions/56860/what-is-the-liskov-substitution-principle>

<http://stackoverflow.com/questions/8161896/example-code-to-show-how-java-synchronized-block-works>

<http://tutorials.jenkov.com/java-concurrency/synchronized.html>

<http://tutorials.jenkov.com/java-util-concurrent/cyclicbarrier.html>

<http://www.dynatrace.com/en/javabook/how-garbage-collection-works.html>

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/G1GettingStarted/index.html>

<http://www.programcreek.com/2013/03/hashmap-vs-treemap-vs-hashtable-vs-linkedhashmap/>

<https://docs.oracle.com/cd/E13150_01/jrockit_jvm/jrockit/geninfo/diagnos/garbage_collect.html>

<https://en.m.wikipedia.org/wiki/Creational_pattern>

<https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)>

<https://lostechies.com/derickbailey/2009/02/11/solid-development-principles-in-motivational-pictures/>

<https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>

For database interviews

<https://www.geeksforgeeks.org/commonly-asked-dbms-interview-questions/>

<https://www.geeksforgeeks.org/commonly-asked-dbms-interview-questions-set-2/amp/?ref=rp>

<https://www.geeksforgeeks.org/sql-interview-questions/amp/?ref=rp>

<https://www.geeksforgeeks.org/sql-interview-questions-set-1/amp/?ref=rp>

<https://www.geeksforgeeks.org/sql-interview-questions-set-2/amp/?ref=rp>

* question on database tuning
* how index internally works
* after partitioning, if query is still taking time, what we can do
* what if a microservice is crashing, what can we do
* what hystrix do apart from fault tolerance
* complete Microservice architecture and spring cloud
* he asked about all the Design patterns. randomly selected few and started asking for example
* Design a component which will have input as order which has product name , quantities and client name as properties  
  output is order summary which is diff for diff product type  
  for ex for stock, stock-quantities  
  for mutual fund, mutualfund-client name  
  how you will design it  
  it should be modifiable + extendable  
    
  2.print all possible combination of balanced parenthesis   
  like if n=2 then (()),  ()()
* caching mechanism. custom implementation of LRU cache
* which header is in rest api that can be used for preventing multiple calls from client
* question abt accept header/ content type  
  after defining accept header, how it works
* what is pipeline continuous integration in Jenkins
* how spring mvc decides which controller to call

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topic** | **Not Proficient** | **Marginally Proficient** | **Proficient** | **Exceeds** |
| **Basics: Mutability, Interface, Equals, Comparable** | Not Covered As it was covered in Round 1 | | | |
| **Exceptions, Java Data Structures** |  | Y |  |  |
| **Threading: sync/lock, executor, thread safety** | Y |  |  |  |
| **Java 8** |  | Y |  |  |
| **Design Pattern and Principals** | Not Covered | | | |
| **Problem Solving** | Y |  |  |  |
| **Edge Case Scenarios** | Y |  |  |  |
| **Database** |  | Y |  |  |
| **Frameworks** | Not Covered | | | |

**Java 8:**

Streams:

1. Explain why and how Java Streams are lazily executed
2. Short Circuiting (Eg: Limit())
3. Stateful (distinct()/sorted()) and Stateless(map()/filter()) intermediate operation
4. Order of execution
5. Which thread pool is used in Parallel streams and how it works
6. Why HashMap value can’t be null if collected in stream

**Concurrency:**

Taxi Aggregator Problem:

 Two opposition party member’s are trying to book Taxi from online App. There can be fight in Cab if allocation does not satisfy below criteria

1. Cab have capacity of 4 passengers
2. Either all the passengers should be of same political party or it should have 2 members from each party
3. Any other combination can lead to fight

**Design:**

Design In Memory file system like Unix with below features but should be extensible

* ls
* mkdir
* appendContent to file
* Read Contents from file

**Database:**

Give table Hacker(hacker\_id, hacker\_name) and Challenge(hacker\_id, challenge\_id)

1. Get total challenges by each hacker(Solved)
2. If 2 hackers have same number of challenges, filter out both(Could not solve)
3. If highest number of challenge are created by multiple hacker, don’t filter out (Could not solve)

**Basics of Core java**

* Exception handling.
  + Checked exception vs unchecked exception
  + Overriding the methods which throws checked or unchecked exceptions
* Generics
* Serialization
* **OOPs concepts**
* Immutable class
* Abstract class and interfaces
* **Encapsulation vs Abstraction**

**Data structures / Java Collections**

* Internals of HashMap, HashSet, TreeMap, TreeSet, LinkedList
* LRU cache implementation (Hint LinkedHashMap)
* Why there is a hash method inside the HashMap class?

(Hint: It’s not for key.hashcode() % bucket\_size)

* Uses of loadfactor and treefy threashold.
* Can HashSet contain a duplicate value?

(Yes, it can when equals and hashcode methods don’t follow the contract)

Find out what is the contract between them?

* Time complexity of all the operations on HashMap, HashSet, TreeMap, TreeSet, Stack and Queue for Best case, worst case and average case.

**SQL, DB2, Stored Procedures**

* Find the most recently hired emps in each department.

<https://www.w3resource.com/sql-exercises/employee-database-exercise/sql-subqueries-exercise-employee-database-52.php>

Follow up:

Can you optimize it further if we are fetching data in batches?

Can you solve it using proprietary functions like row\_number()

* Practice joins

<https://www.w3resource.com/sql/joins/sql-joins.php>

* Write a query in SQL to list the details of the departments where maximum number of employees are working.
* Write a query in SQL to find one employee from each department who is getting the highest salary within his department. (Don’t forget to handle corner case where multiple / no employees are getting highest salary in a department.)
* Write a query in SQL to list the details of the departments where maximum number of employees are working. (1 department with max employee or multiple departments having same number of employees)

**Microservices using spring webmvc, spring boot**

* **What is the use of service registry and discovery?**

* **Get familiar with usage of zookeeper, eureka and consul as a service discovery. Added advantage if you know trade off of all these solutions.**
* What are the key benefits of using Spring Boot?
  + Auto Configuration
  + Dependency management using starter parent
  + Actuator and Health Checks
  + Embedded Web Server
* How to implement Fault tolerance and circuit breaker?
* CAP theorem.
* What is the role of API Gateway in microservice architecture?
* How do you decide that a microservice should expose HTTP or JMS listener?
* **Microservices design patterns specially SAGA patterns, API Composition pattern and CQRS pattern**

**Spring Framework**

* AOP concepts like Join point, Advice etc.

**Problem Solving**

1. Implement a queue using stacks. Prepare for follow up questions also.
2. Find first and last positions of an element in a sorted array. (Observe the pattern like here array is sorted and it may contain duplicates)

<https://www.geeksforgeeks.org/find-first-and-last-positions-of-an-element-in-a-sorted-array/>

1. Detect and Remove Loop in a Linked List.

<https://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/>

1. Add two numbers represented by linked lists.
2. Inorder Tree Traversal without Recursion.
3. How to design an LRU cache. Which data structure you would like to use?

Hint: LinkedHashMap Or HashMap + LinkedList

1. Find the k most frequent words from a file.

<https://www.geeksforgeeks.org/find-the-k-most-frequent-words-from-a-file/>

1. Print N’th node from the end of a Linked List.

<https://www.geeksforgeeks.org/nth-node-from-the-end-of-a-linked-list/>

1. Find K largest (or smallest) elements in an array.

There are multiple ways to solve this but using Min/max heap is the better solution.

<https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/>

**Algorithms**

* Tim sort (Used by Java for objects)
* Quick Sort (Double Pivot Quick Sort is used by Java for primitives)
* Merge sort

**Multithreading / Concurrency**

* **Intrinsic and extrinsic locks**
* **How wait() and notify() works internally?**
* **Spurious wakeup and SpinLock**
* **Synchronization puzzle like print 1 to N sequentially from 2 threads one should be printing even numbers and one should be printing odd numbers.**
* **Deadlock prevention techniques**
* When we should prefer volatile int/long over the Atomic Integer / Atomic Long?
* When we should prefer Atomic Integer / Atomic Long over the intrinsic and extrinsic locks?
* How to Design a website visitor counter. Come up with at least 3 solutions and be aware about the pros and cons of each one.
* **Get Familiar with Semaphores, CountDownLatch and CyclicBarrier.**

<https://www.baeldung.com/java-semaphore>

<https://www.baeldung.com/java-cyclicbarrier-countdownlatch>

* Assume you have to replace the thread pool of any web application server using your custom thread pool executor. Try to build it and submit some tasks to it.

Do load testing of the same.

<https://howtodoinjava.com/java/multi-threading/java-thread-pool-executor-example/>

* Get familiar with class level and object level locks.

<https://howtodoinjava.com/java/multi-threading/object-vs-class-level-locking/>

**Design Principles / Patterns**

* SOLID Design Principles

Real world use cases, good to relate it with the work you have done

* Singleton Design pattern

Java vs Spring IOC, how to break it?

* Strategy Design Pattern
* Decorator Design Pattern
* Adaptor Design Pattern
* Builder Design Pattern
* Observer Design Pattern
* Template Design Pattern

**JVM Internals**

* **Diagnose Out of memory error**.
* Heap dumps, memory dumps

1. difference between jre and jdk
2. steps to execute code without eclipse
3. how is class loaded by class loader
4. internal working of Thread local
5. create a design structure for snake and leader program
6. how can class execute without jdk

1.

import java.util.Collection; import java.util.List; public interface Library { // Add a new book into the library void addBook(int bookId, String section); // Move the book to another section . A // book can be in only one section void updateBook(int bookId, String section); // Return the collection of ids of all the books // belonging to the specified section Collection<Integer> listBooks(String section); }   
  
2.

You can assume areaUnderCurve is a monotonically increasing function of "X". You need to write a function that will return the point coordinate where the area under the curve gets divided exactly in half. double findHalfAreaPoint(double X)

e.g. if areaUnderCurve(X) returns 100, then findHalfAreaPoint(X) should return a value "H" such that areaUnderCurve(H) will return 50.

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. Ways of using final keyword in java.

Can constructor be final?

If a final primitive variable is not initialized, can that be assigned some value?

How to make list unmodifiable if not final?

2. Overloaded methods with primitive variables and another with wrapper variables. Which one gets called? How is ambiguity resolved?

3. Given a list of integers, find the max product using any 3 values without repetition.

4. Hashmap implementation. Problem with using inefficient hashcode computation.

**For any other query feel free to reach me**

**My questions.**

* 1. What is externalization? Serializable vs Externalizable.
  2. Is java pass by value or pass by reference?
  3. What is n+1 problem in hibernate? What are isolation levels in hibernate?
  4. Strong reference vs weak reference in Java.
  5. What are marker interface? Why do we need them.
  6. Fail fast and fail safe iterator in java.
  7. Can we call static method from null object?
  8. Get unique elements from hashmap by using stream.
  9. @Transactional for transaction management in spring. Propagation and Isolation level.
  10. @Controller vs @RestController
  11. Inheritance vs composition
  12. Immutable list vs Collections.unmodifieableList.
  13. Observer design pattern in Java.
  14. Abstract class vs Interface in Java 8.
  15. Abstract class vs Functional interface in java.
  16. What is sql injection? How to prevent sql injection?
  17. How to integrate log4j with hibernate?
  18. Can we run single spring boot application on multiple port?
  19. What are embedded application server supported in spring boot?
  20. Prototype design pattern in Java.