Question Bank

Java Questions -

1. Explain about BigDecimal and how is it different from 'double' with the use cases about where to use it.
2. Pass By Value / Pass By Reference in Java?
3. When should we use a LinkedList vs ArrayList?
4. What is the time complexity of an ArrayList and LinkedList?
5. What is the time complexity of addition of an item to an Arraylist, Hashmap and LinkedList?
6. What is the time complexity - if we add an element at beginning of an ArrayList? Also In case of LinkedList?
7. What is time complexity - O(n) to find the length of Linked List?
8. How is HashSet implemented – Internal working?
9. How HashMap is implemented – Internal working?
10. How does Get and Put work in an HasMap?
11. I have a hash map, how do I sort it based on values? e.g. I have a hash map, of employee => DOB, how to find out people who were born in March.
12. What is the purpose of the hashCode() method? When would you need to override it?
13. What is the relationship between hashCode() and equals()?
14. What is a hash collision?
15. How to avoid a hash collision?
16. When to use a HashMap vs ConcurrentHashMap?
17. What is the difference between using a HashMap in a synchronized block vs ConcurrentHashMap?
18. What is meant by fail safe and fail fast iterators ?
19. Will Adding or Removing elements from a non-concurrent collection cause an Exception? If yes, what Exception.
20. How does insertion into a TreeMap work?
21. What is the structure of JVM? What are the different components within the architecture? (garbage collection, heap, etc, where does variables go, static variables go, etc.)
22. What is PermGen space, Heap space
23. Types of garbage collectors available in Java.
24. How does Garbage Collection work?
25. Features of Java 9,10, 11
26. How to make a class immutable?
27. Memory Management in Java - What does a Memory leak happen in Java? Will this cause memory leak - public static List<Double> list = new ArrayList<>(); ?
28. What does the static keyword mean?
29. How are Static Blocks initialized in Java?
30. What is the difference between final, finally, finalize?
31. What is Big(O) notation ?

## Concurrency

1. Any experience on Multi-threading? What happens if a thread is started again (Illegal thread exception)
2. What is the concurrency level (default 16)
3. Happens-Before relationship
   * <https://www.logicbig.com/tutorials/core-java-tutorial/java-multi-threading/happens-before.html>
   * <https://docs.oracle.com/javase/tutorial/essential/concurrency/memconsist.html>
4. Happens-After relationship
5. What is volatile – use cases – Why volatile is called volatile?
   * <https://www.baeldung.com/java-volatile>
6. What is the difference between == and .equals?
8. What is the difference between an abstract class and an interface?
9. What is the difference between checked and unchecked exceptions and how would you handle them?
10. SOLID principles what are they?
11. what's problem in below code  
    @PostMapping  
      
    public void trades (Trade trade) {  
      
    saveToDB(trade);  
      
    publishToService(trade);  
      
    }

## Hibernate

1. Define eager and lazy fetch types and which one to use for one to many relationships
2. Relational DB – expect some basic questions like how indexing works

# Microservices

1. How does a Microservice differ from SOA?
2. How would you create a Springboot project?
3. What is an Actuator? What is it used for?
4. What is Environment in Spring boot?
   * <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/core/env/Environment.html>
5. What is Transactional in Spring – Typical scenario – A method having Transactional annotation – what would happen 2 threads are spawned from it. – How transaction means here in case of threads
   * <https://dzone.com/articles/how-does-spring-transactional>
6. How to check the health of spring boot application
7. What are the metrics you are concerned about from Actuator?

# Devops

1. CI CD pipeline – How will I implement – what is the thought process?
   * <https://dzone.com/articles/learn-how-to-setup-a-cicd-pipeline-from-scratch>
2. What are steps for Jenkins setup from new projects?
3. What are the ways to perform CI/CD?
4. How is a Jenkins pipeline built?
5. How you deploy in prod - Git , CI / CD pipeline

# Cloud

1. IaaS vs PaaS vs SaaS
2. How is migration to cloud planned

# AWS questions

1. How do we scale up a service on AWS
2. How is a Docker image deployed to AWS EC2?
3. How are objects in S3 stored and what is the consistency model it follows.
4. How is the health check of an application performed when it is deployed in ECS? How to know if an application is working fine?
5. How does an application load balancer distribute the load across instances?
6. How are different environments managed in AWS?

# Messaging

1. Kafka and Other messaging tools – MAF/Mule – what is the advantage?
2. What will happen in the case of Kafka and other messaging – if a message is sent and retrieved from consumer
3. Kafka related questions, like partitioning, how to guarantee the correct sequence of messages is received by the consumer
4. Kafka basics how do you make sure the messages delivered in order
5. Kafka listener

# Containers and Orchestration

1. How to scale up/down in Kubernetes
2. How does Kubernetes manage failovers
3. How does deployment work in Kubernetes
4. What is a service in Kubernetes?
5. File  Structure in Docker
6. How is a Docker image built?
7. What is the first instruction on a Docker file?
8. Spring Boot fat jar - ENTRYPOINT

# Code share question bank

## Algorithms

1. convert a number to literal (e.g. 23 should be converted to twenty-three)
2. reverse a number without using strings
3. Find Clock Angle
4. Find if the string is an anagram
5. Find some window moving average
6. Rotate matrix by 90 degrees
7. Programme to solve the [1, 3, 4, -3, 2] if you start jumping based on the value array element in how many steps, can come out. Negative numbers go back.
8. Write a pseudo code for strategy pattern.
9. Write a pseudo code for TreeMap implementation.
10. Questions on binary tree
11. Complexity of treemap implementation.
12. Find the next index in Load Balancer which is using Round Robin Technique

/\*RoundRobinLoadBalancer\*/  
  
   
  
Class Loadbalancer{  
  
 public static int calculateNextIndexPositionToBeAssigned{  
  
/\*get the next index in a RoundRobinLoadbalancer to put an instance into\*/    
  
  
  
}

## Fill in gaps programs

1. Implement Redis Cache
2. Implement load balancer

Others

public List<String> findClosestAddresses(File addresses, String hotelAddres) {};

The problem to solve was, how would you read the file and write the solution to find closest addresses to the given address. No API's given nothing, you basically can use whatever you want, I believe they want to see how you approach this problem. Again depends on which approach you choose and you and interviewer will decide on the file structure and how to extract the distance. Once you agree on this you start implementing the business logic for returning the List<String> of closest addresses