CITI (03-Feb-2021) (1 hour 30 min around)

Java -

1. What are the OOPS concept? And their example?
2. Overriding vs overloading?
3. Is it possible to widen access modifiers in overriding? What about exception?
4. Why Runtime exception can be thrown from overridden method but not broader checked exception?
5. Can we override static methods? No – Why?
6. Where to use array list vs linked list?
7. How does HashMap internally work?
8. What happens when hash collision occurs?
9. If my HashMap is working slow, what could be the reason?
10. What is Semaphore?
11. What is Executors Framework? How does it work?
12. How synchronized block works? (monitors)? Class level and object level locks?
13. What design pattern you used?
14. What is Façade design pattern?
15. Should the HashMap key object be immutable? Why?
16. How to make a class immutable?
17. If one class say Employee has account list, then how to make the employee class immutable?
18. Abstract class vs interface? Where to use one over another?
19. Can abstract class have constructor? What is the use of it if we cannot instantiate it?

Spark -

1. How do you get SparkContext/Session in your program?

**SparkConf** sparkConf = new SparkConf().setAppName("Word Count Demo").setMaster("local");

**JavaSparkContext** jsc = **new** JavaSparkContext(sparkConf);

**SparkSession** spark = SparkSession

.*builder*()

.appName("Java Spark SQL basic example")

.config("spark.master", "local")

.getOrCreate();

SparkContext sc = spark.sparkContext();

What is SparkContext? - SparkContext (aka **Spark context**) is the entry point to Spark for a Spark application. It [sets up internal services](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext-creating-instance-internals.html) and establishes a connection to a [Spark execution environment (deployment mode)](https://mallikarjuna_g.gitbooks.io/spark/content/spark-deployment-environments.html).

Once a [SparkContext instance is created](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext.html" \l "creating-instance) you can use it to [create RDDs](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext.html#creating-rdds), [accumulators](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext.html#creating-accumulators) and [broadcast variables](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext.html#broadcast), access Spark services and [run jobs](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext.html#runJob) (until SparkContext is [stopped](https://mallikarjuna_g.gitbooks.io/spark/content/spark-sparkcontext.html#stop)).

A Spark context is essentially a client of Spark’s execution environment and acts as the master of your Spark application (don’t get confused with the other meaning of [Master](https://mallikarjuna_g.gitbooks.io/spark/content/spark-master.html) in Spark, though).

1. What all are the context used in your current spark job? (spark,sql,hive context etc)
2. Where does spark conf reside? (file path of spark-default.conf ?)

The default Spark properties file is [$SPARK\_HOME/conf/spark-defaults.conf](https://mallikarjuna_g.gitbooks.io/spark/content/spark-properties.html#spark-defaults-conf) that could be overriden using spark-submit's [--properties-file command-line option](https://mallikarjuna_g.gitbooks.io/spark/content/spark-submit.html#properties-file).

<https://mallikarjuna_g.gitbooks.io/spark/content/spark-properties.html>

1. What all are the properties in that file?

spark.submit.deployMode setting can be client or cluster

# Example:

# spark.master spark://master:7077

# spark.eventLog.enabled true

# spark.eventLog.dir hdfs://namenode:8021/directory

# spark.serializer org.apache.spark.serializer.KryoSerializer

# spark.driver.memory 5g

# spark.executor.extraJavaOptions -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"

1. What is your cluster configurations?
2. How much your current Aggregation job takes time?
3. What is executor’s memory used in your cluster?
4. What is RDD? What is Dataframe and Dataset?
5. To create DataFrame don’t need to define object type? – No
6. How does Spark fast then MapReduce? - In memory
7. Which resource manager you are using? Why did you use YARN ? – yarn is better suited for Hadoop cluster, and if you already have hadoop cluster then yarn is better choice.