CITI (08-Feb-2021) (2nd round 1 hour around)

Java -

1. If I need to maintain insertion order in map how can I do that? – LinkedHashMap
2. How LinkedHashMap work internally?
3. How TreeMap work internally? – it is red black tree – self balanced tree?
4. What is red-black tree?
5. Which GC you used? What is G1 and CMS GC?

The heap is partitioned into a set of equal-sized heap regions, each a contiguous range of virtual memory. Certain region sets are assigned the same roles (eden, survivor, old) as in the older collectors, but there is not a fixed size for them. This provides greater flexibility in memory usage.

1. What are the OOPS concept? Do you think these are useful?
2. What is encapsulation?
3. What is aggregation, composition association? - Association is relation between two separate classes which establishes through their Objects. Association can be one-to-one, one-to-many, many-to-one, many-to-many.  
   In Object-Oriented programming, an Object communicates to other Object to use functionality and services provided by that object. **Composition** and **Aggregation** are the two forms of association.

Aggregation : It is a special form of Association where:

* It represents **Has-A** relationship.
* It is a **unidirectional association** i.e. a one way relationship. For example, department can have students but vice versa is not possible and thus unidirectional in nature.
* In Aggregation,**both the entries can survive individually** which means ending one entity will not effect the other entity.

Composition : It is a restricted form of Aggregation in which two entities are highly dependent on each other.

* It represents **part-of** relationship.
* In composition, both the entities are dependent on each other.
* When there is a composition between two entities, the composed object **cannot exist** without the other entity.

**Aggregation vs Composition**

1. **Dependency:** Aggregation implies a relationship where the child **can exist independently** of the parent. For example, Bank and Employee, delete the Bank and the Employee still exist. whereas Composition implies a relationship where the child **cannot exist independent** of the parent. Example: Human and heart, heart don’t exist separate to a Human
2. **Type of Relationship:** Aggregation relation is **“has-a”** and composition is **“part-of”** relation.
3. **Type of association:**Composition is a **strong** Association whereas Aggregation is a **weak** Association.
4. Print odd even in two threads but in a sequence –
5. The wait and notify on which you called? Means why these methods are in objects and not in thread?
6. Why Synchronized/locking is required?
7. Threadlocal variables when to use?
8. Cache thread pool vs Fixed thread pool?
9. How many ways threads can be created?
10. How many threads one can create? How threads are created internally?
11. What is RejectedExecutionException? When do we get and how to handle?

New tasks submitted in method execute(java.lang.Runnable) will be rejected when the Executor has been shut down, and also when the Executor uses finite bounds for both maximum threads and work queue capacity, and is saturated. In either case, the execute method invokes the RejectedExecutionHandler.rejectedExecution(java.lang.Runnable, java.util.concurrent.ThreadPoolExecutor) method of its RejectedExecutionHandler. Four predefined handler policies are provided:

In the default ThreadPoolExecutor.AbortPolicy, the handler throws a runtime RejectedExecutionException upon rejection.

In ThreadPoolExecutor.CallerRunsPolicy, the thread that invokes execute itself runs the task. This provides a simple feedback control mechanism that will slow down the rate that new tasks are submitted.

In ThreadPoolExecutor.DiscardPolicy, a task that cannot be executed is simply dropped.

In ThreadPoolExecutor.DiscardOldestPolicy, if the executor is not shut down, the task at the head of the work queue is dropped, and then execution is retried (which can fail again, causing this to be repeated.)

It is possible to define and use other kinds of RejectedExecutionHandler classes. Doing so requires some care especially when policies are designed to work only under particular capacity or queuing policies.

1. Design pattern which one you have used? – Observer, Façade, Factory, singleton
2. If I have a complex service and I want to simplify then which design pattern should I use? – Façade.
3. How to find cycle in a linked list? – using two pointer
4. What is the space complexity in merge and quick sort?
5. How to create queue like functionality from stack? – using two stack, but the time complexity will be O(n) for get
6. What are the ACID properties in DB? Atomicity, consistency, Isolation and Durability **Atomicity**  
   By this, we mean that either the entire transaction takes place at once or doesn’t happen at all.

**Consistency :**  
This means that integrity constraints must be maintained so that the database is consistent before and after the transaction.

**Isolation**  
This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of database state.

**Durability:**  
This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs.

1. Spring version?
2. Spring Interceptor and filter?