**Use Cases**

1. What will be your design in the scenario where 250 MB of memory is available to you and 10 file of 100 MB coming from upstream the requirement is to sort these 10 file and save it in database
2. Three threads t1,t2,t3. t1 prints 1,4,7.. And t2 pring 2,5,8..and t3 prints 3,6,9….

Make them to print 1,2,3,4,5,6,7,8,9… in a sequence and then repeat after 9.

1. How do you implement producer consumer   problem where there are ten producers and ten consumers.
2. Implement Producer Consumer problem using blocking queue / using wait and notify
3. Implement Thread pool
4. Implement cyclic brier/Countdown Latch
5. Implement Merge Sort
6. Design a concurrent Counter
7. Design a generic  object pool
8. Design an multithreaded  Download Manager which can show the progress of different downloads
9. Design a concurrent HashMap  using lock splitting concept
10. Implement the parallel sort using Cyclic Barrier.
11. Sort hash map by values
12. Write program to find largest and second largest element in an unsorted array
13. Design a Call centre

Imagine you have a call centre with three levels of employees: fresher, technical lead (TL), product manager (PM). There can be multiple employees, but only one TL or PM. An incoming telephone call must be allocated to a fresher who is free. If a fresher can’t handle the call, he or she must escalate the call to technical lead. If the TL is not free or not able to handle it, then the call should be escalated to PM. Design the classes and data structures for this problem

1. Create spring based service that exposes primarily two features:

* First time registration of the user.
* Validation of the existing user credentials.
* You have to use the dependency injection feature of spring framework (Spring Annotations and configuration).

1. Create a data structure which stores Order based on Price and Security Id.

* User should be able to get all orders for a given Price or all orders for a Security or all order for a particular price and security.
* Multiple threads will try to insert, delete or search the orders in the data structure.
* (Hint: use concurrent hash map).
* Implement the version by using ConcurrentSkipListMap<K,V> for better performance.

1. In System, we are storing security master data which is used to build order or to watch stock.

* Storing huge data in-memory is creating issues for the system.
* Use Hibernate Caching to fetch and store the security data in Cache.
* Also update the market price of stock in cache itself.
* Write JUnit to measure the performance of the application. (Hint :Demonstrate the Hibernate Performance metrics cache)

1. Create a POJO User with fields UserName, First Name, Last Name, Creation Date, Password, and Active (boolean).

Create a mapping POJO and map the properties to the table column.

Create a DAO which can do following action

* + Register a user
  + Authenticate User
  + Reset password

Write Unit Test to verify the behavior of DAO

1. Setup Active MQ and create 2 queues - >orderQueue and executedOrderQueue.

* Write Spring JMS code to send/listen/receive message
* Demonstrate different Acknowledgement types
* Demonstrate Message Converter to change XML message into Order Object.
* Setup marketDataTopic as a Topic and Write at least 5 clients connected to topic and getting market Data. Market Data POJO would be Id, Security Code, Price, change.