Interview Questions

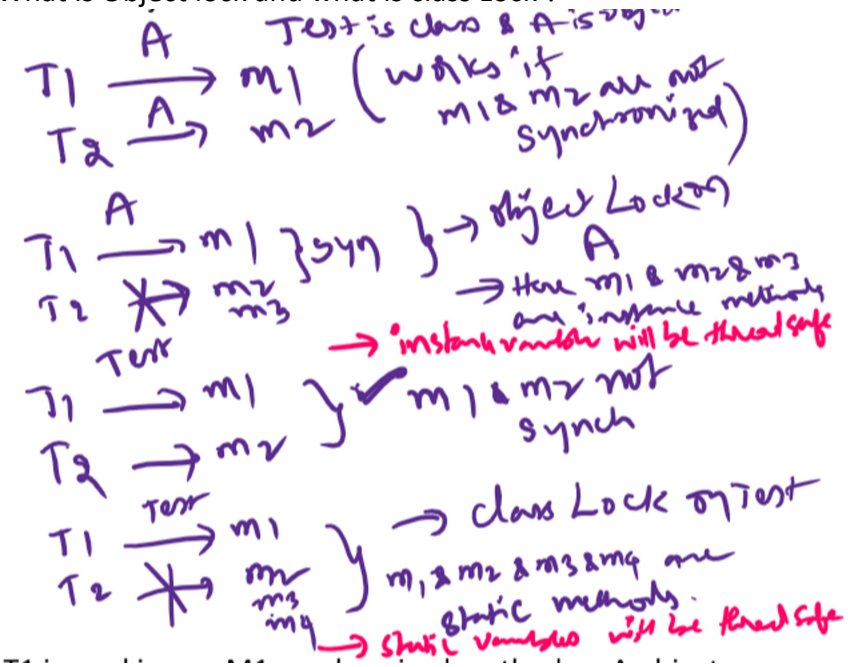
- What is Thread?
  - It is an independent path of execution with in a program.
- How many ways we can create the thread Object ?
  - 3 ways -> extending Thread, implementing Runnable, implementing callable
- What are the different states in the Thread ?
  - New, runnable, running, dead, blocked.
- What happens when we call start on dead thread ?
  - Illegal thread state exception -> by using variable called thread status. First time when we call start method thread status value is 0. and it gets set to some non zero value after starting the independent path. So next time when we call start the thread status variable value is non zero and it throws exception
- Can we call start method on a thread twice ?
  - No->same answer as above
- Where the thread flow of execution start ?
  - Run method or call method
- How to set priority of a thread ?
  - SetPriortity();1- MIN, 10 MAX PR- 5 NORM. main thread priority is 5
- How can we set a name to Thread ?
  - setName or Thread constructor
- What is join method of a thread ?
  - Join method makes sure that thread can be continued only after a specific thread is completed

```
Main (){
T1.start()
T1.join -> Main thread will wait for t1 to complete
Sysout();
}
Main(){
Thread t= new Thread(new Runnable{
Public void run(){
        Create some other thread->
        Thread t2= new Thread(new ThreadExample())
        T2.start()
        T2.join() -> here T thread will wait for t2 thread to complete the work
}
});
T.start();


Class ThreadExample Implements runnable{
}
```

- What is the difference between sleep and wait ?

- Refer to the notes

- What is inter thread communication ?
  - One thread talking to other thread is called interthread communication.
  - T1 -> A-> m1 synchronized method -> acquires object Lock on A and if we have wait method inside synchronized block -> release lock
  - T2- >A->m2 synchronized method -> because of Object lock on A T1 will wait and it will execute only after wait method has released the lock.-> wait then t2 will go to wait mode
  - T3 ->B->m3 synchronized method -> acquires object Lock on B->
  - T4 -> A -> m5 synchronized method ->when it says notify -> any one of T1 or T2 will comeback to execution where notify all -> T1 and t2 both will come back to execution. T3 will not comeback because it has lock on B object and not A
- Why wait notify and notify all are present in Object class ?
  - Refer to notes -> Lock is acquired on Object and not on thread and not on method. One thread should tell another thread to come back to execution. This is only possible if the method are in object class because object is parent for all.
- What is Synchronization ?
  - T1 -> A-> m1 method
  - T2 ->A -> m2 - no access
  - Using synchronized keyword
- What is Object lock and what is class Lock ?
- 

- T1 is working on M1 synchronized method on A object
  Can the T2 execute M2 synchronized method on A object ?
  - No because of Object lock on A
- What is the disadvantage of general way of creating the thread ?
  - Thread cannot be restarted or re used.
- What kind of data can we make thread safe using Class lock ?
  - Static data or static variables of class
- What is Thread pool ?
  - Pool of threads. Or collection of thread. To over come the disadvantage of general thread object creation

- How many thread pools are present in java?
  - 4->refer to notes
- How the thread will take the work in thread pool
  - Internally the thread pool uses the queue. And all the thread will take the work or task from the queue.
- Why are we going for callable interface ?
  - From runnable we cannot return the value. Using callable a value can be returned from thread
- What is the abstract method present in callable ?
  - Call
- Is thread platform dependent or platform independent ?
  - Platform dependent
- What is the difference between a thread and a Runnable interface in Java?
  - By Extending Thread we cannot perform multiple inheritance because using extends keyword we can extend only 1 class
  - Implementing the interface will make segregation of duties. We just need to override the run method. Thread behavior will be taken care by Thread class so implementing the Runnable will be easy and good.
- What is Semaphore ?
  - Allows permitted thread to execute the shared resource
  - In general way. If we create 5 threads and working 5 different object-> 5 threads will start executing on 5 object and work parallelly but I want to limit to only 2 to threads .
- What is a race condition, and how can you prevent it in Java?
  - Refer to notes
- What is join method and how it is different from count down latch ?
  - Join method makes sure that single thread is completed
  - Count down latch makes sure that all the threads in a pool have completed the executing all the tasks present in the queue.
- What is daemon thread ? How to create daemon thread ?
  - Background execution thread is called as daemon thread -> example: Garbage collector-> system.gc()
  - setDaemon() method to be called on user defined thread to make it as background execution thread.
- What is a Cyclic Barrier, and how does it work?
  - Cyclic barrier will make all the threads to come to a point and then execute the rest of the logic. Refer to notes for more information
- What is a volatile variable, and why is it useful in multithreading?
  - Volatile will resolve visibility issues. Once thread cannot see what is happening in another thread
  - Suppose if both the threads are using the same variable then the value may not be updated properly because thread are independent path of execution.
  - Refer notes for more information
  - Volatile is applicable for variable
- What is a Thread Local variable, and how does it work?
  - Refer to notes: it is a local variable for thread
- What is the Fork/Join framework in Java
  - Refer to Notes
- What is the difference between an optimistic and pessimistic lock?
  - Refer to Notes

- How can you interrupt a running thread in Java?
  - By calling the interrupt. Immediately it will not stop the execution. You can check whether the interrupt method is called or not by calling the isInterrupted() method
- What are the differences between Synchronized vs Lock ?
  - Acquire the lock across method is not possible in synchronization
  - Fair lock nd unfair lock is also not possible in synchronization
  - tryLock functionality is not present in synchronization
- What is Thread Groups ?
  - Thread group is used to group the threads. Refer notes

Programmatic questions on Thread

- Write a program to create and start a new thread in Java.
- Write a program to create a thread using the Runnable interface in Java.
  - Thread t= new Thread(object of runnable interface that means object of class which implements runnable interface)
- Write a program to demonstrate the concept of synchronization in Java.
  - Refer to com.ashokit.synchronize package in the code
- Write a program to prevent a race condition in Java.
  - Same as synchronization.
- Write a program to demonstrate the use of the wait and notify methods in Java.
  - Refer to com.ashokit.threads.classlock code
- Write a program to demonstrate the concept of deadlock in Java.
  - com.ashokit.deadlock code
- Write a program to implement a producer-consumer problem using threads in Java
  - com.ashokit.producerandconsumer code