Q:- what is synchronization.?
Ans:- page 309 pdf notes
=================================
Q:- synchronized block in recursive method what will be happen.?
===
Q:- Thread life Cycle .?
Ans:- from pdf notes page 324
=======================================
Q:- object and class level locking .?
Ans:-

Object Level Lock:-
=================

Internally synchronization concept is implemented by using lock concept.
Every object in java has a unique lock. Whenever we are using synchronized keyword then only lock concept will come into the picture.
If a Thread wants to execute any synchronized method on the given object 1st it has to get the lock of that object. Once a Thread got the lock of that object then it's allow to execute any synchronized method on that object. If the synchronized method execution completes then automatically Thread releases lock.
While a Thread executing any synchronized method the remaining Threads are not allowed execute any synchronized method on that object simultaneously. But remaining Threads are allowed to execute any non-synchronized method simultaneously. [lock concept is implemented based on object but not based on method].

Class Level Lock:-
=================

1. Every class in java has a unique lock. If a Thread wants to execute a static synchronized method then it required class level lock.
2. Once a Thread got class level lock then it is allow to execute any static synchronized method of that class.
3. While a Thread executing any static synchronized method the remaining Threads are not allow to execute any static synchronized method of that class simultaneously.
4. But remaining Threads are allowed to execute normal synchronized methods, normal static methods, and normal instance methods simultaneously.
5. Class level lock and object lock both are different and there is no relationship between these two.
===
sleep and wait usage how it works.?
Ans:- page 306 and 314
==
synchronized block.?
ANs:- 312
==========================

deadloack .?
Ans:- 319
==
synchronization lead to deadlock
==
which synchronization is better block VS method.?
Ans:- blck bcz in bloack we can synchronize only particular line which we want ot be synchronized.
==
draw back of threads.
Ans:- 1. Writing multithreaded programs requires very careful design.
      2. Debugging a multithreaded program is much, much harder than debugging a single-threaded one,
                because the interactions between the threads are very hard to control.
==============================
Q:- synchronization.? how to avoid deadlock.?
ANs:- It’s a very complex process and not easy to catch. But still if we try, we can avoid this. There are some methods by which
      we can avoid this condition. We can’t completely remove its possibility but we can reduce.

Avoid Nested Locks : This is the main reason for dead lock. Dead Lock mainly happens when we give locks to multiple threads.
                 Avoid giving lock to multiple threads if we already have given to one.
Avoid Unnecessary Locks : We should have lock only those members which are required. Having lock on unnecessarily can lead to dead lock.
Using thread join : Dead lock condition appears when one thread is waiting other to finish. If this condition occurs we can use
                 Thread.join with maximum time you think the execution will take.
==
Q:- Thread pool .? how it work.?
Ans:- page 344
=====================================
dirty thread.?
======================
Q:- how to create thread.? why there are two ways to create thread.?
Ans page 287
==
Q:-Is servlet thread safe by default.?
Ans:-By default, servlets are not thread-safe
==
Q:- how you can make them thread safe.?
Ans:- Servlet is not thread-safe by itself. You can make it thread-safe by
making the service method synchronized. you need to implement SingleThreadInterface to

make it thread-safe.
=====================================
diff b/w notify vs notifyALL method?
Ans: -page 317
==
Object are thread safe or memeber variables? how to make a variabke thread safe.?
Ans:- declare a variable volatile for thread safe.?
===
how we can find thread is dead or alive.?
Ans:- getState() and IsAlive();
==
After completion of life cycle what is the state of a thread .?
Ans:- Dead or terminated.
==
thread communication
Ans :- page 314
=========================
if thread A is active and B,C,D are in running state then how we can start C.?
==
======
Q:- how to run thread.?
Ans :- t.start();
====================
Q:- implementation of start.?
Ans:- This method registers this thread with a thread scheduler.
Performs all other mandatory activities, like to create a new child thread other than previous thread or main thread.
finally, It invokes the run() method which consists task of this thread.
================================
if we execute run method on a class/thread then what will be happen.?
Ans: nothing will be happen bcz of empty implementation in that method
=================================
Q:- wait, notify, notifyAll() methods are thread class method why all are defined in object class.?
ANS:- wait(), notify() and notifyAll() methods are available in Object class but not in
Thread class because Thread can call these methods on any common object.
==
diff b/w runnable interface and thread class.
ANs:- page 298
===
Diff b/w t.start() and t.run() method.?
Ans:--   In the case of t.start() a new Thread will be created which is responsible for the execution of run() method.
 But in the case of t.run() no new Thread will be created and run() method will be

executed just like a normal method by the main Thread.