Synchronized: is the modifier applicable only for blocks and methods but not for classes and variables.

If multiple threads try to operate simultaneously on the same java object then there may be chance of data inconsistency problem,

To overcome this problem we should go for synchronized keyword.

If a method or block declare as synchronized then at a time only one thread is allow to execute that method or block on the given object.

so that data-inconsistency problem will be resolved.

Advantage: The main advantage of synchronized keyword is we can resolve data-inconsistency problems.

But the main Disadvantage of synchronized is :it increases waiting time of threads and creates Performance problems.

Hence : If there is no specific requirement it is not recommended to use synchronized keyword.

--------------------------------------------------------END------------------------------------------------------------------------------

Internally Synchronization concept is implemented by using Lock.

Every Object in JAVA has a unique lock.

Whenever we are using synchronized keyword then only lock concept come into the pictures.

t1----------> lock(x)

Synchronize m1(){

-------------

-------------

-------------

-------------

}

If a thread wants to execute synchronized method on the given object first it has to get lock of that object.

Once thread got the lock then it is allow to execute any synchronized method on that object.

Once method execution completes automatically thread releases lock.

Note: Acquiring and releasing lock internally takes care by JVM, and programmer is not responsible for this activity.

Very Important Note: While a thread executing synchronized method on a given object that remaining threads not allow to execute any synchronized method simultaneously

on the same object. but remaining thread are allow to execute non-synchronized method simultaneously.

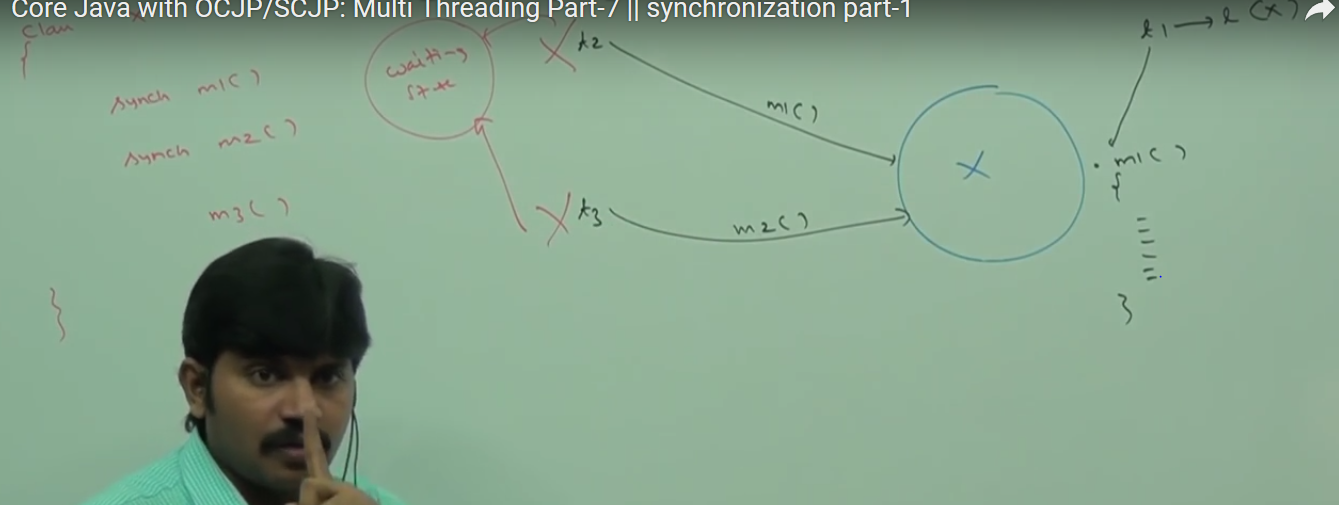
Example:

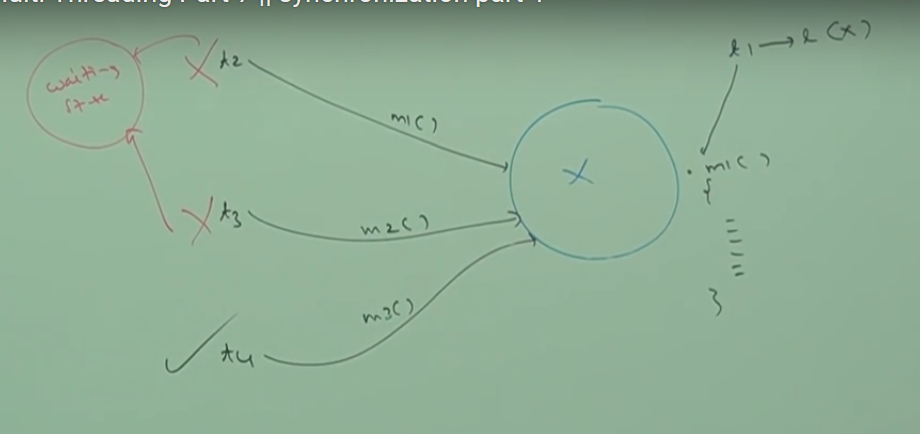
Class x{  
synchronized m1()

synchronized m2()

m3()

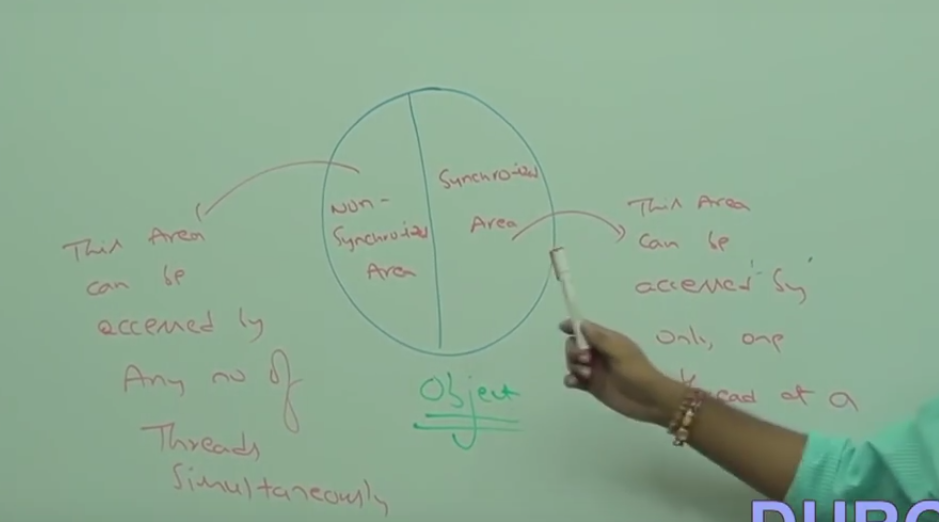
}





Lock concept is implemented based on the object but not based on method.

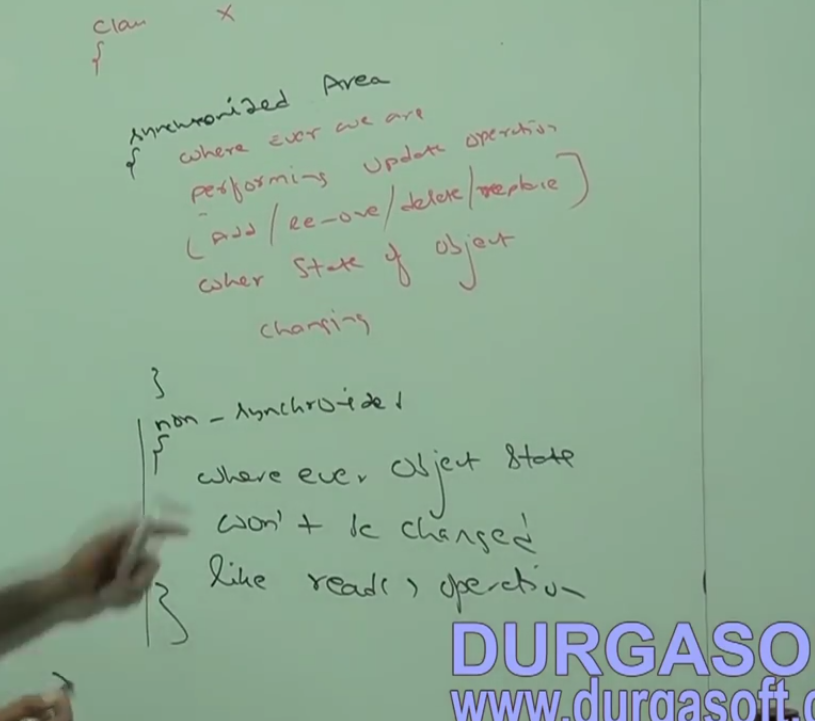
Refer Below diagram

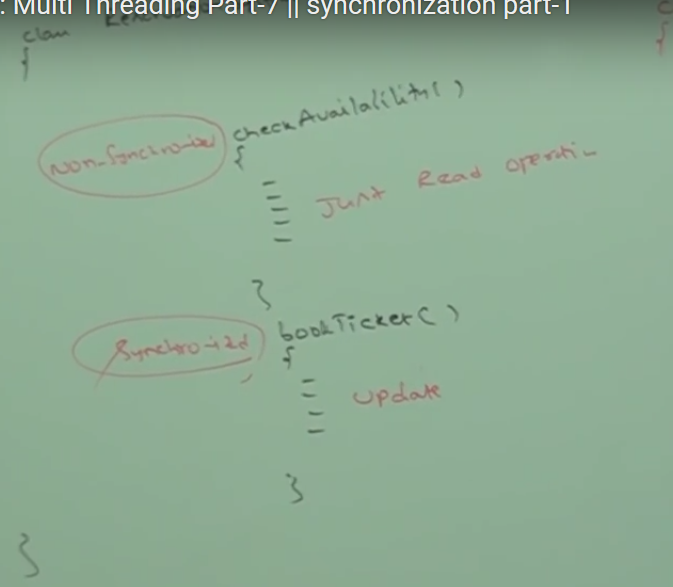


Object

Synchronized: This area can be accessed one thread at a time

Refer Below Diagram imp

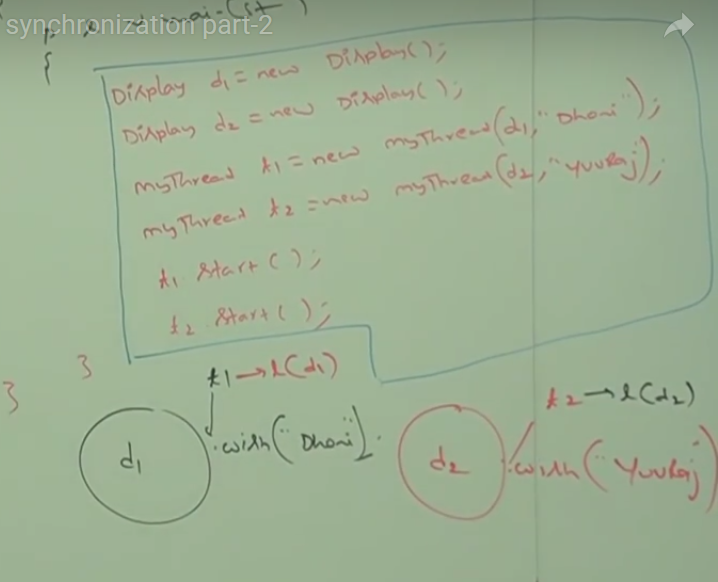




--------------------------------------------------------------------------------------------------------------------------------------

**Case Study:**

Even though wish method is synchronized we will get irregular output because threads are operating on different java object.



Conclusion: if multiple threads are operating on same java object, then synchronization is required.

If multiple threads are operating on multiple java objects then synchronization is not required.

Class Level Lock:

Every class in java has a unique lock which is nothing but class level lock.

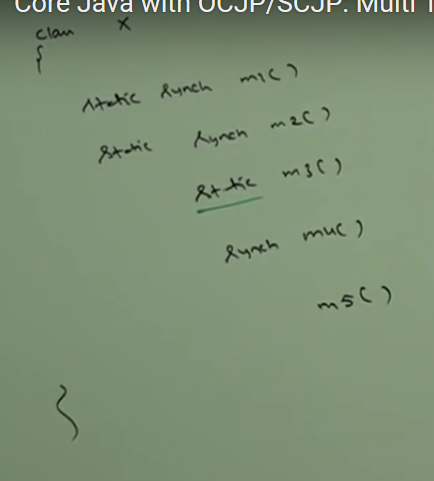
If a thread wants to execute **static synchronized method** then thread required class level lock.  
Once thread got class level lock then it is allow to execute any static synchronized method of that class.  
Once method execution completes automatically thread releases the lock.

Static Synchronized Method: it will be class-level ock.

While a thread executing static synchronized method the remaining thread not allow to execute any static synchronized method of that simultaneously but remaining thread are allow to execute the following method simultaneously

1: normal static methods.  
2: Synchronized instance methods.   
3: Normal method.

Screen short for same

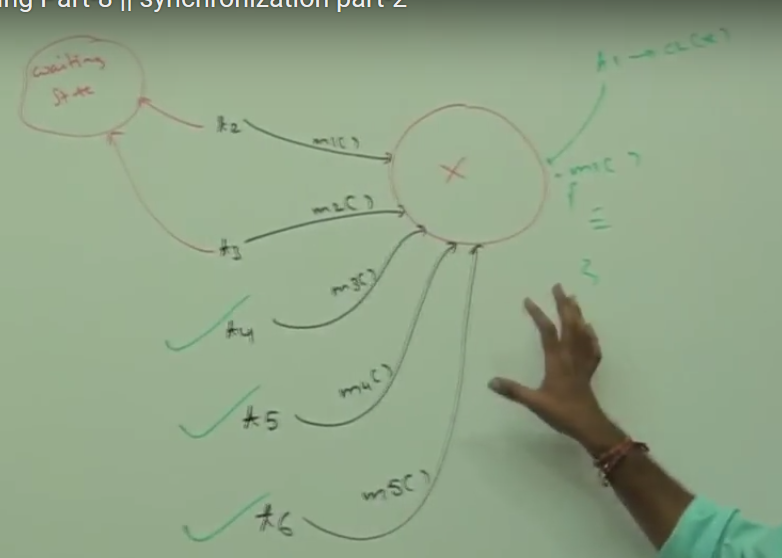
  
Class x{

static synch m1()

static synch m2()

static m3()

synch m4()  
 m5()  
}



Synchronized Block:

If very few line of code required to synchronization then it is not recommended to declare entire method as synchronized.  
we have to enclose those few lines of code by using synchronized block.  
  
The main advantage of synchronized block on over synchronized method it reduces waiting time of threads and improve performance of system/application.