**What is a Deadlock ?**

The term **Deadlock** refers to a situation in multithreading, where two or more threads are blocked forever, waiting for each other.  
It is a simply a condition where a set of threads are blocked because each thread is holding a resource and waiting for another resource acquired by some other thread.  
Since, all the threads are waiting for each other to release the resource, the condition is called Deadlock.

we use **synchronized** keyword to make the class or method thread-safe.The keyword ensures that only one thread can access the resource at a given point of time. A synchronized block is synchronized on some object. The block ensures that only one thread executes inside it at a time. All other threads attempting to enter it are blocked until the thread inside it exits.

If we use two synchronized block might cause deadlock

Example of deadlock:

If we created tow constant string variables resourse1, resourse2 to use them to produce or consume , in run function we created objects of two threads t1, t2 and use two synchronized block on resourse1, resourse2

For producer

**Thread t1 = new Thread() {**

synchronized ( resource 1 )

\*\*/

\* Do some work

\*/

synchronized ( resource 2 )

{

For consumer

***Thread t2 = new Thread() {***

*synchronized ( resource 2 )*

\*\*/

\* Do some work

\*/

synchronized ( resource 1 )

{

what is happening here –

1. Thread 1 starts and acquires lock on resource 1.
2. Thread 2 starts and acquires lock on resource 2.
3. Thread 1 prints "Thread 1: I have locked Resource 1" and Thread 2 prints "Thread 2: I have locked Resource 2 ".
4. Thread 1 tries to take object lock of resource 2 but it is already acquired by Thread 2 so it waits till it become free. It will not release lock of Resource 1 until it gets lock of Resource 2.
5. And same happens with thread 2. It tries to take object lock of resource 1 but it is already acquired by Thread 1 so it waits till it become free. It will not release lock of Resource 2 until it gets lock of Resource 1.
6. Now, both threads are in wait state, and are waiting for each other to release the locks.
7. As none of the thread is ready to release lock, so this is the Dead Lock condition.
8. And hence when we run the program, it seems as if the program execution has stopped.