# CoreJava Multi-Threading

Important keywords used in Java multi-threading programming.

* Thread class
* Runnable interface

Q. How to create Threads ?  
Ans - There are 2 ways to create thread,

* By extending Thread class
* By implementing Runnable interface

Q. What happens when you start a Thread twice ?  
Ans - We get IllegalThreadStateException when we start a Thread twice.

Q. Explain join() method.  
Ans - The join() method waits for a thread to die. In other words, it causes the currently running threads to stop executing until the thread it joins with completes its task.

Important notes about variables in context of Thread :-

* Local variables are stored in Thread's stack
* However any object which we create using new keyword is stored on heap
* So anything stored in heap is shared across thread.
* And anything stored in Thread's stack is not shared across stack

Loook at these words - Thread interference, Race Condition

Q. What is a Synchronized keyword in Java ?  
Ans - A Java synchronized block marks a method or a block of code as synchronized. Java synchronized blocks can be used to avoid race conditions.  
Synchronized blocks in Java are marked with the synchronized keyword. A synchronized block in Java is synchronized on some object.  
All other threads attempting to enter the synchronized block are blocked until the thread inside the synchronized block exits the block.

The synchronized keyword can be used to mark four different types of blocks:

Instance methods  
Static methods  
Code blocks inside instance methods  
Code blocks inside static methods

Q. What do we mean by "Synchronized blocks in Java are re-entrant?  
Ans - Synchronized blocks in Java are reentrant. This means, that if a Java thread enters a synchronized block of code, and therelby take the lock on the monitor object the block is synchronized on, the thread can enter other Java code blocks synchronized on the same monitor object.