ThreadB that calls the join() on ThreadA  
Then ThreadB will have to wait until ThreadA completes its execution

When ThreadA finishes, ThreadB will resume its execution

[*https://raw.githubusercontent.com/vsaravanan/java22/master/src/main/java/com/saravanjs/java22/console/multithreading/ThreadJoin.java*](https://raw.githubusercontent.com/vsaravanan/java22/master/src/main/java/com/saravanjs/java22/console/multithreading/ThreadJoin.java)

*public class* ThreadJoin {  
 *public static void* main(String[] args) {  
 Thread threadA = *new* Thread(() -> {  
 System.*out*.println("ThreadA started");  
 *try* {  
 System.*out*.println("Sleeping at A");  
 Thread.*sleep*(2000); *// Simulating some work* } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println("ThreadA completed");  
 });  
  
 Thread threadB = *new* Thread(() -> {  
 System.*out*.println("ThreadB started");  
 *try* {  
 threadA.start();  
 System.*out*.println("ThreadA joined with ThreadB");  
 threadA.join();  
 } *catch* (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println("ThreadB completed");  
 });  
  
  
 threadB.start();  
  
 System.*out*.println("Main thread completed");  
 }  
}

Main thread completed

ThreadB started

ThreadA joined with ThreadB

ThreadA started

Sleeping at A

ThreadA completed

ThreadB completed

In case if you don’t join, then both threads will be executed in parallel  
ThreadB could complete before the Thread A as below  
  
  
Main thread completed

ThreadB started

ThreadA joined with ThreadB

ThreadA started

Sleeping at A

ThreadB completed

ThreadA completed