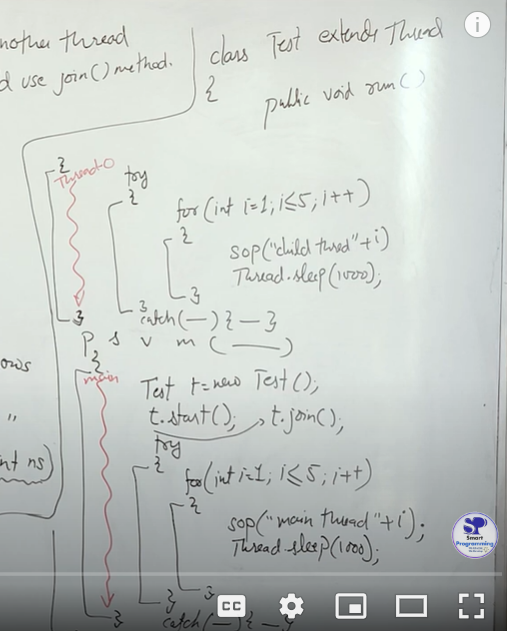


SEE THE DIAGRAM AND LEARN

Like test drive thread needs to wait for medical thread in order to achieve driving license . similarly officer thread needs to wait for driving license for final pass.

Throws InterruptedExceptions - try,catch ; exceptions ; InterruptedExceptions can be used to handle.

l

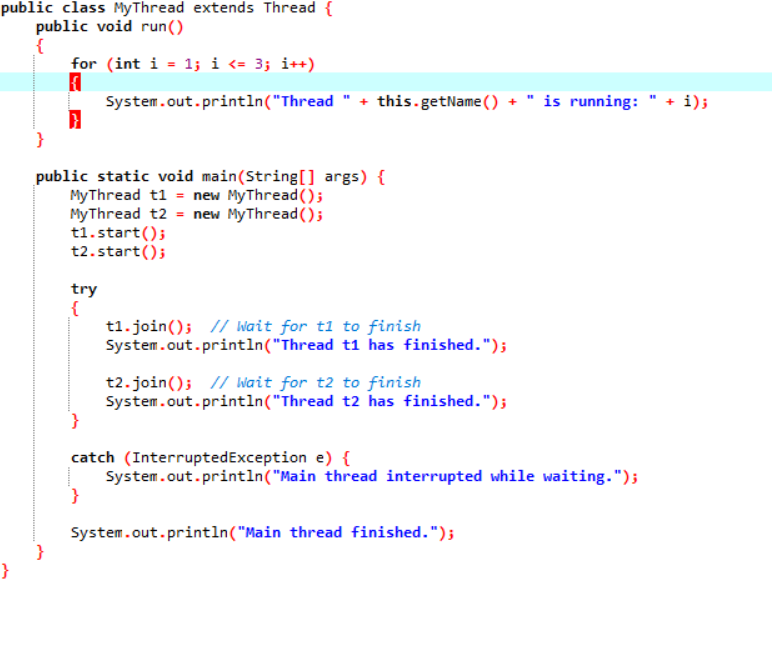
Like inside main method we are using t.join() so first thread of t will be executed then main method try catch block.

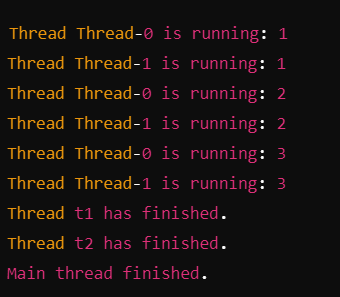
so try catch need to be used in both main and t1 ??

No, you **only need to use the try-catch block** in the thread where you call the \*\*join()\*\* method, which is usually in the **main thread** or any other thread that is waiting for another thread to complete.

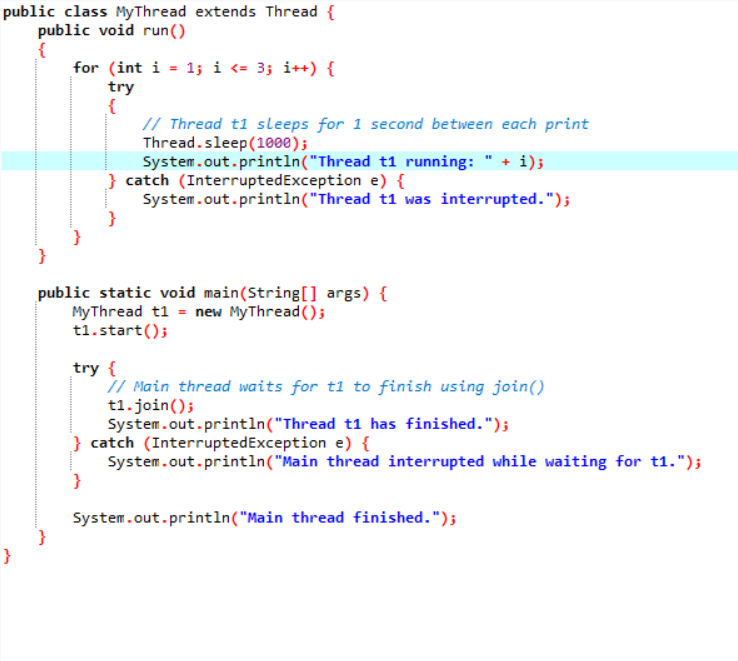
You **do not need** a try-catch block in the thread that is being "joined" (e.g., t1), unless that thread (like t1) contains other operations that throw exceptions (such as Thread.sleep()).

E.g this will work fine >>

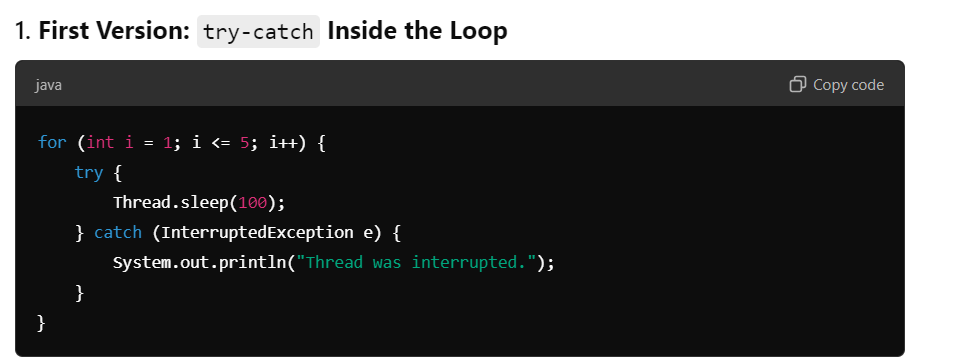




This too work fine since sleep method used , try catch used inside run()

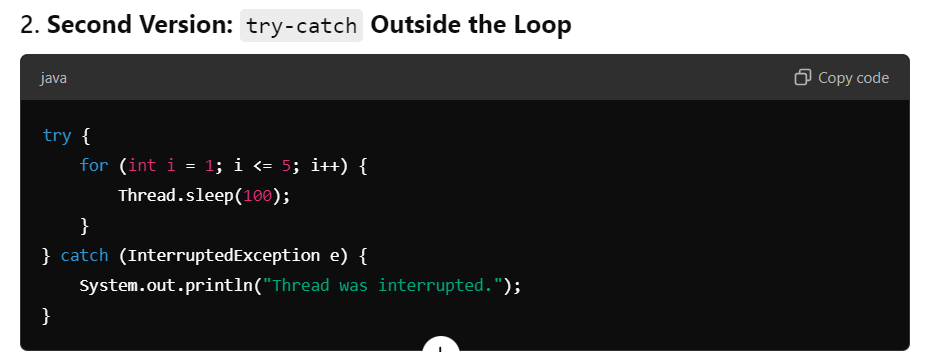


q> RANDOM – HOW TO USE TRY METHOD??



 If the thread is interrupted during one sleep period, it catches the exception but continues to run the loop for subsequent iterations.

 This is useful if you want the loop to continue after an interruption.

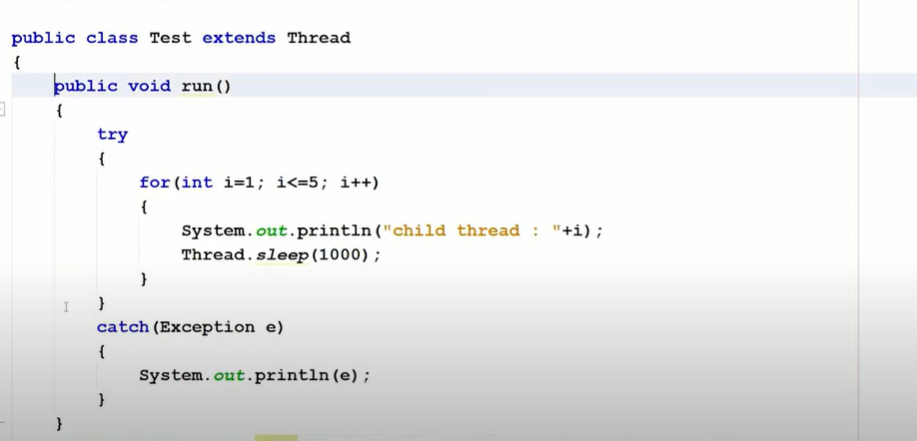


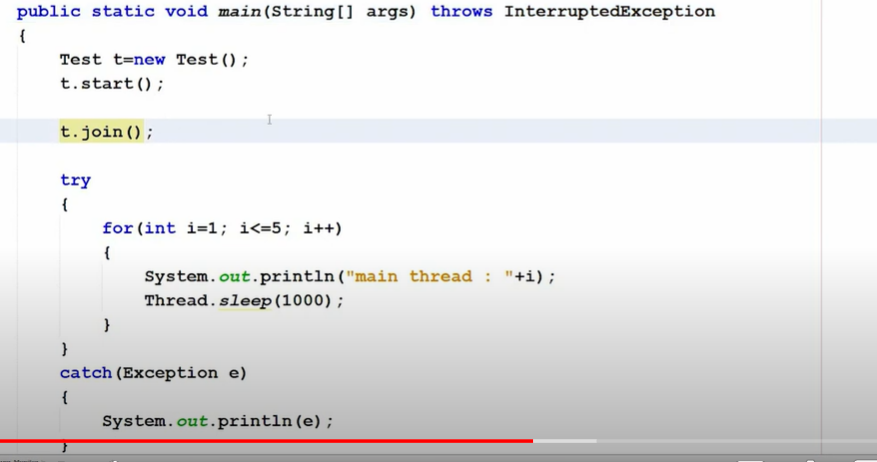
 If the thread is interrupted, it stops the entire loop. This might be desired if you don't want the loop to continue after an interruption.

 There’s only one try-catch block, making the code slightly cleaner.

So we just need to put thread methods like thread.sleep() inside try,catch block . Other code can be anywhere.

A>>

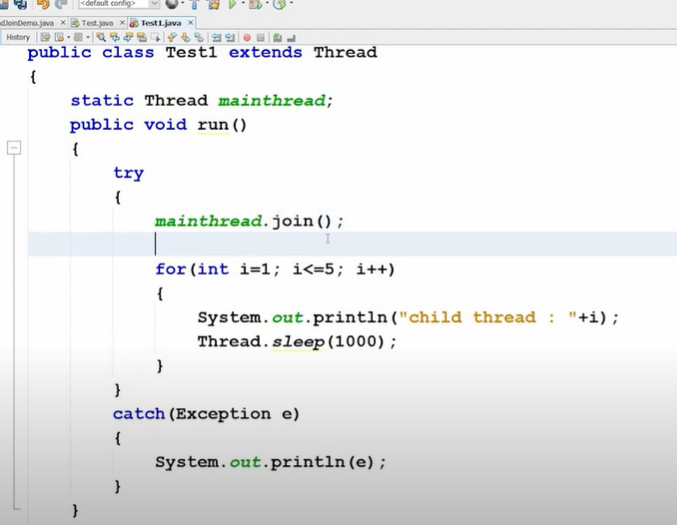


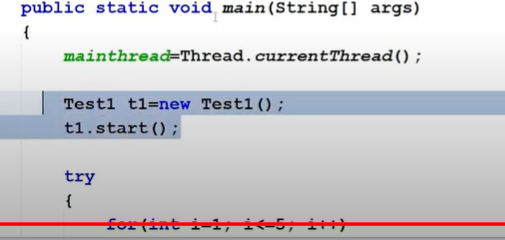


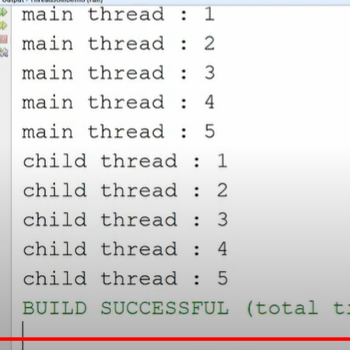


t.join() inside main thread means main method waits till run method is not executed.

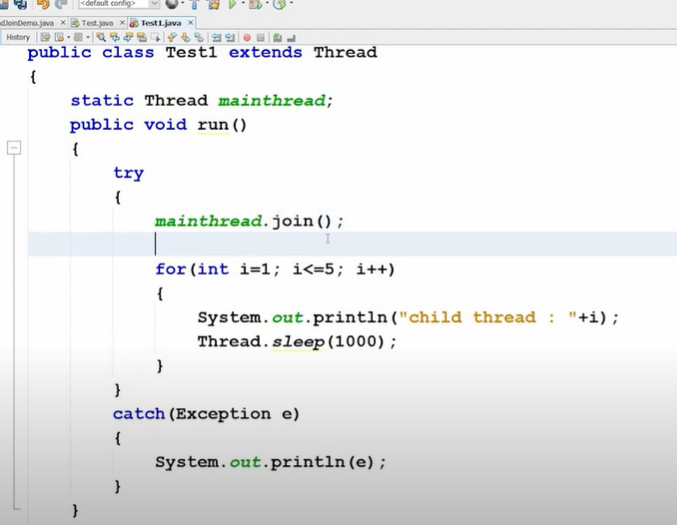
B>>similarly to make run method wait we need to use main\_thread.sleep() inside run method . How to get mainthread ?-> using current thread which pass reference !!







C>>Why static ? ( CODE B )



 Now, mainthread is an **instance variable**, so each instance of Test1 will have its own mainthread reference. In the main() method, we must manually assign the current thread (Thread.currentThread()) to t1.mainthread.

 This means each Test1 instance would have its own copy of mainthread, and we would need to set it separately for each instance. If you were to create multiple Test1 instances, each one could have a different reference, which could cause confusion or incorrect behavior.

 **Compilation Error**: If you remove static, you'll get a compile-time error in the run() method, stating that you cannot reference a non-static field mainthread from a static context.

 **Accessing the Main Thread**: Without static, each instance of Test1 would have its own mainthread reference, which isn't what you want. The goal here is to reference the **main thread of the program**, and since the main method is static, mainthread must also be static to be accessed within the main method.

So how to access mainthread using non static variable??

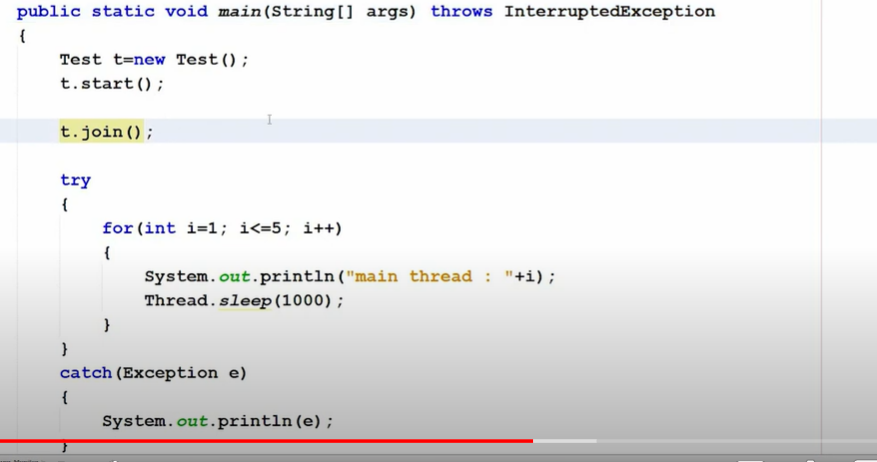
->

Test1 t1 = new Test1();

t1.mainthread = Thread.currentThread(); // We must manually assign the main

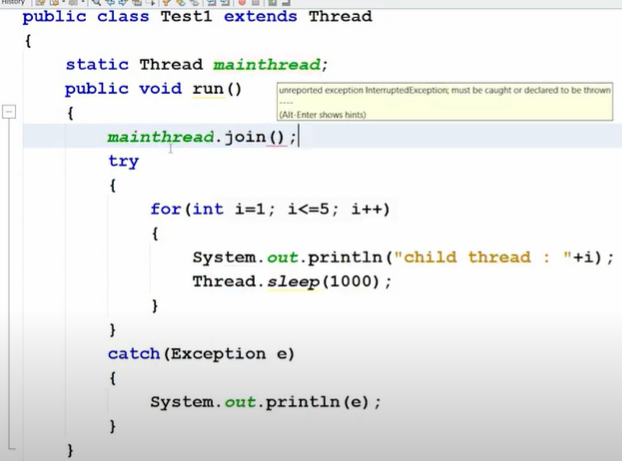
thread t1.start();

q>

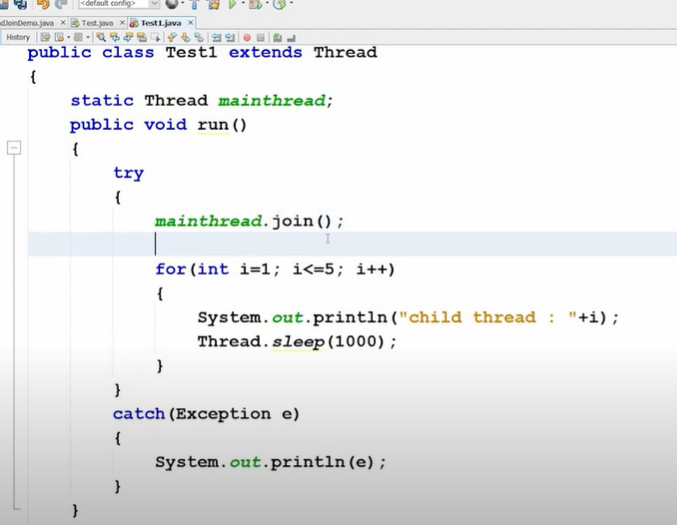


We need to use Interrpted Exception or try catch block enclosing

t.join().



Example : This code will give error since mainthread.join() is not enclosed within try catch block or exception not handled.



This one will correctly handle exception!

q> Example :

