***join() method:***

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

### Syntax:

public void join()throws InterruptedException

public void join(long milliseconds)throws InterruptedException

***InterruptedException:***

An **InterruptedException** is thrown when a thread that is sleeping, waiting, or is occupied is interrupted.

***EXAMPLE:***

***class TestJoinMethod1 extends Thread{***

***public void run(){***

***for(int i=1;i<=5;i++){***

***try{***

***Thread.sleep(500);***

***}catch(Exception e){System.out.println(e);}***

***System.out.println(i);***

***}***

***}***

***public static void main(String args[]){***

***TestJoinMethod1 t1=new TestJoinMethod1();***

***TestJoinMethod1 t2=new TestJoinMethod1();***

***TestJoinMethod1 t3=new TestJoinMethod1();***

***t1.start();***

***try{***

***t1.join();***

***}catch(Exception e){System.out.println(e);}***

***t2.start();***

***t3.start();***

***}***

***}***

***OUTPUT:***

1

2

3

4

5

1

1

2

2

3

3

4

4

5

5

As you can see in the above example,when t1 completes its task then t2 and t3 starts executing.

***Example of join(long miliseconds) method:***

***class TestJoinMethod2 extends Thread{***

***public void run(){***

***for(int i=1;i<=5;i++){***

***try{***

***Thread.sleep(500);***

***}catch(Exception e){System.out.println(e);}***

***System.out.println(i);***

***}***

***}***

***public static void main(String args[]){***

***TestJoinMethod2 t1=new TestJoinMethod2();***

***TestJoinMethod2 t2=new TestJoinMethod2();***

***TestJoinMethod2 t3=new TestJoinMethod2();***

***t1.start();***

***try{***

***t1.join(1500);***

***}catch(Exception e){System.out.println(e);}***

***t2.start();***

***t3.start();***

***}***

***}***

**OUTPUT:**

1

2

3

1

4

1

2

5

2

3

3

4

4

5

5

In the above example,when t1 is completes its task for 1500 miliseconds(3 times) then t2 and t3 starts executing.