// Java program to demonstrate thread states

class thread implements Runnable {

public void run()

{

// moving thread2 to timed waiting state

try {

Thread.sleep(1500);

}

catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println(

"State of thread1 while it called join() method on thread2 -"

+ Test.thread1.getState());

try {

Thread.sleep(200);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

public class Test implements Runnable {

public static Thread thread1;

public static Test obj;

public static void main(String[] args)

{

obj = new Test();

thread1 = new Thread(obj);

// thread1 created and is currently in the NEW

// state.

System.out.println(

"State of thread1 after creating it - "

+ thread1.getState());

thread1.start();

// thread1 moved to Runnable state

System.out.println(

"State of thread1 after calling .start() method on it - "

+ thread1.getState());

}

public void run()

{

thread myThread = new thread();

Thread thread2 = new Thread(myThread);

// thread1 created and is currently in the NEW

// state.

System.out.println(

"State of thread2 after creating it - "

+ thread2.getState());

thread2.start();

// thread2 moved to Runnable state

System.out.println(

"State of thread2 after calling .start() method on it - "

+ thread2.getState());

// moving thread1 to timed waiting state

try {

// moving thread1 to timed waiting state

Thread.sleep(200);

}

catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println(

"State of thread2 after calling .sleep() method on it - "

+ thread2.getState());

try {

// waiting for thread2 to die

thread2.join();

}

catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println(

"State of thread2 when it has finished it's execution - "

+ thread2.getState());

}

}