**//Example 2: Demonstration\_2.java**

/\* Use of suspend() and resume() methods \*/

class Thread1 extends Thread {

public void run( ) {

try{

System.out.println (" First thread starts running" );

sleep(10000);

System.out.println (" First thread finishes running" );

}

catch(Exception e){ }

}

}

class Thread2 extends Thread {

public void run( ) {

try{

System.out.println ( "Second thread starts running");

System.out.println ( "Second thread is suspended itself ");

**suspend( );**

System.out.println (" Second thread runs again" );

}

catch(Exception e){ }

}

}

class Demonstration\_5{

public static void main (String args[ ] ){

try{

Thread1 first = new Thread1( ); // It is a newborn thread i.e. in Newborn state

Thread2 second= new Thread2( ); // another new born thread

first.start( ); // first is scheduled for running

second.start( ); // second is scheduled for running

System.out.println("Revive the second thread" ); // If it is suspended

**second.resume( );**

System.out.println ("Second thread went for 10 seconds sleep " );

second.sleep (10000);

System.out.println ("Wake up second thread and finishes running" );

System.out.println ( " Demonstration is finished ");

}

catch(Exception e){ }

}

}

**Output:**

Revive the second thread

First thread starts running

Second thread starts running

Second thread is suspended itself

Second thread went for 10 seconds sleep

**//Example 3: Demonstration\_3.java**

import java.io.\*;

// Creating thread by creating the objects of that class

class ThreadJoining extends Thread

{

@Override

public void run()

{

for (int i = 0; i < 2; i++)

{

try

{

Thread.sleep(500);

System.out.println("Current Thread: "

+ Thread.currentThread().getName());

}

catch(Exception ex)

{

System.out.println("Exception has" +

" been caught" + ex);

}

System.out.println(i);

}

}

}

class **Demonstration\_3**

{

public static void main (String[] args)

{

// creating two threads

ThreadJoining t1 = new ThreadJoining();

ThreadJoining t2 = new ThreadJoining();

ThreadJoining t3 = new ThreadJoining();

// thread t1 starts

t1.start();

**// starts second thread after when first thread t1 has died.**

try

{

System.out.println("Current Thread: "

+ Thread.currentThread().getName());

**t1.join();**

}

catch(Exception ex)

{

System.out.println("Exception has " +

"been caught" + ex);

}

// t2 starts

t2.start();

**// starts t3 after when thread t2 has died.**

try

{

System.out.println("Current Thread: "

+ Thread.currentThread().getName());

**t2.join();**

}

catch(Exception ex)

{

System.out.println("Exception has been" +

" caught" + ex);

}

t3.start();

}

}

**Output:**

Current Thread: main

Current Thread: Thread-0

0

Current Thread: Thread-0

1

Current Thread: main

Current Thread: Thread-1

0

Current Thread: Thread-1

1

Current Thread: Thread-2

0

Current Thread: Thread-2

1