class Shared

{

synchronized void test1(Shared s2)

{

System.out.println("test1-begin");

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

}

s2.test2(this);

System.out.println("test1-end");

}

synchronized void test2(Shared s1)

{

System.out.println("test2-begin");

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

}

s1.test1(this);

System.out.println("test2-end");

}

}

class Thread1 extends Thread

{

private Shared s1;

private Shared s2;

public Thread1(Shared s1, Shared s2)

{

this.s1 = s1;

this.s2 = s2;

}

public void run()

{

s1.test1(s2);

}

}

class Thread2 extends Thread

{

private Shared s1;

private Shared s2;

public Thread2(Shared s1, Shared s2)

{

this.s1 = s1;

this.s2 = s2;

}

public void run()

{

s2.test2(s1);

}

}

public class dlock

{

public static void main(String[] args)

{

Shared s1 = new Shared();

Shared s2 = new Shared();

Thread1 t1 = new Thread1(s1, s2);

t1.start();

Thread2 t2 = new Thread2(s1, s2);

t2.start();

try

{

Thread.sleep(100);

}

catch (InterruptedException e)

{

}

}

}