

Name: Sayyed Sohail Rashid	Course Name: DC-LAB
Class: BE-CO	Batch: 01
Roll no: 18CO48	Experiment No: 06

Aim : To Implement the Deadlock Detection Algorithm.

Code:

TestThread.java

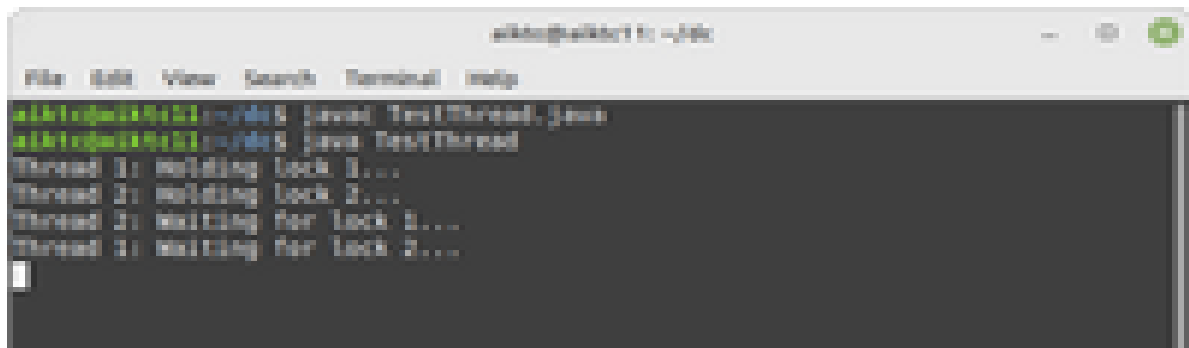
```
import java.io.*;
public class TestThread {
    public static Object Lock1 = new Object();
    public static Object Lock2 = new Object();

    public static void main(String args[]){
        ThreadDemo1 T1 = new ThreadDemo1();
        ThreadDemo2 T2 = new ThreadDemo2();
        T1.start();
        T2.start();
    }

    private static class ThreadDemo1 extends Thread {
        public void run() {
            synchronized (Lock1) {
                System.out.println("Thread 1: Holding lock 1...");
                try{ Thread.sleep(10); }
                catch(InterruptedException e){}
                System.out.println("Thread 1: Waiting for lock 2...");
                synchronized(Lock2){
                    System.out.println("Thread 1: Holding lock 1 & 2...");
                }
            }
        }
    }

    private static class ThreadDemo2 extends Thread {
        public void run() {
            synchronized(Lock1){
                System.out.println("Thread 2: Holding lock 1 & 2...");
            }
        }
    }
}
```

Output:

A screenshot of a terminal window titled 'alishah@alishah: ~'. The terminal shows the execution of a Java program. The first two lines are the command to compile and run 'TestThread.java'. The subsequent four lines show the output of the program, indicating that Thread 1 is waiting for lock 2, Thread 2 is waiting for lock 1, and both threads are waiting for the other to finish, resulting in a deadlock. The terminal text is as follows:

```
alishah@alishah: ~$ javac TestThread.java
alishah@alishah: ~$ java TestThread
Thread 1: waiting lock 2...
Thread 2: waiting lock 1...
Thread 1: waiting for lock 2...
Thread 2: waiting for lock 1...

```

Conclusion:

Deadlock has been successfully detected between the two threads.