

Name: Sayyed Sohail Rashid	Course Name: DC-LAB
Class: BE-CO	Batch: 01
Roll no: 18CO48	Experiment No: 05

Aim : To Implement the Deadlock Detection Algorithm.

Code:

TestThread.java

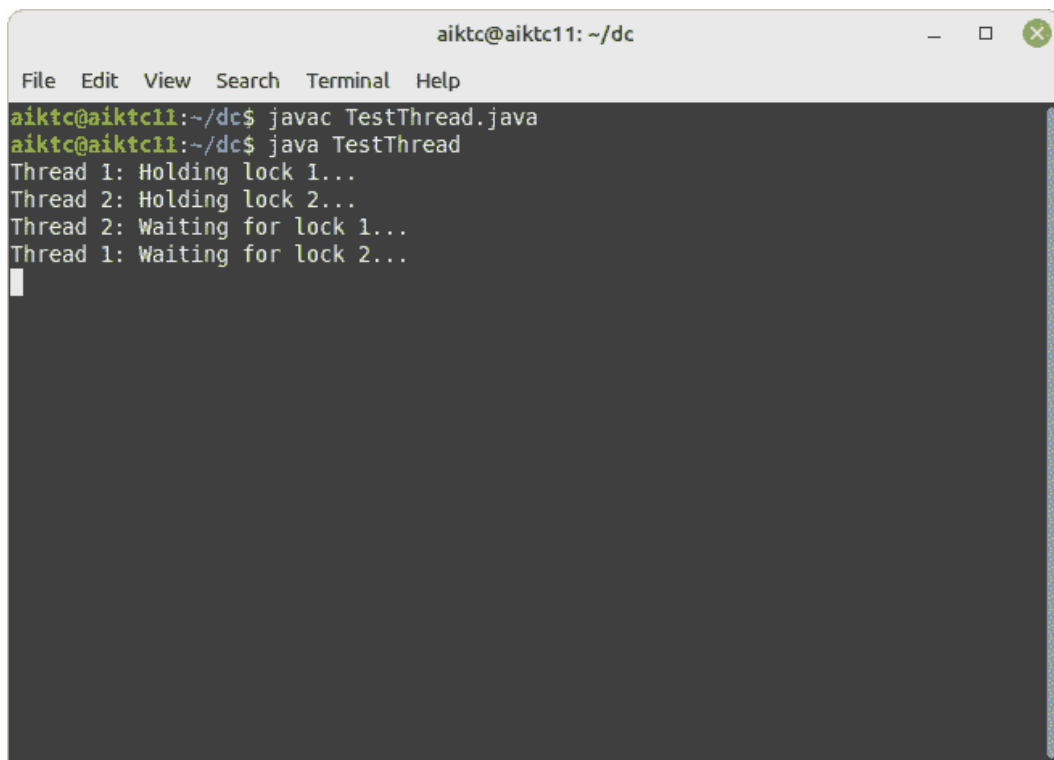
```
import java.io.*;
public class TestThread {
    public static Object Lock1 = new Object();
    public static Object Lock2 = new Object();

    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...");
                }
            }
        }
    }

    synchronized(Lock1){
        System.out.println("Thread 2: Holding lock 1 & 2...");
    }
}
```

Output:

A terminal window titled 'aikt@aiktc11: ~/dc' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'javac TestThread.java' and 'java TestThread'. The output indicates a deadlock state where Thread 1 holds lock 1 and Thread 2 holds lock 2, with each thread waiting for the other to release its lock.

```
aikt@aiktc11:~/dc$ javac TestThread.java
aikt@aiktc11:~/dc$ java TestThread
Thread 1: Holding lock 1...
Thread 2: Holding lock 2...
Thread 2: Waiting for lock 1...
Thread 1: Waiting for lock 2...
```

Conclusion:

Deadlock has been successfully detected between the two threads.