

# LAB 6

## Deadlock in concurrency

**Objective :** Implementing multiple thread blocked resources with help of lock and deadlock conditions.

**Task #1:** Create three threads by implementing thread synchronization block through 3 locks.

```

1 package lab;
2 class Task1 extends Thread {
3     Object A, B, C;
4     Task1(Object A, Object B, Object C) { this.A=A; this.B=B; this.C=C; }
5     public void run() {
6         synchronized(A) {
7             System.out.println("Abdullah acquired Lock A");
8             try { Thread.sleep(10); } catch(Exception e){}
9         synchronized(B) {
10            System.out.println("Abdullah acquired Lock B");
11            try { Thread.sleep(10); } catch(Exception e){}
12        synchronized(C) {
13            System.out.println("Abdullah acquired Lock C");
14            System.out.println("Abdullah released all Locks\n");
15        }
16    }
17 }
18 }
19 }
20 class Task2 extends Thread {
21     Object A, B, C;
22     Task2(Object A, Object B, Object C) { this.A=A; this.B=B; this.C=C; }
23     public void run() {
24         synchronized(A) {
25             System.out.println("038 acquired Lock A");
26             try { Thread.sleep(10); } catch(Exception e){}
27         synchronized(B) {
28             System.out.println("038 acquired Lock B");
29             try { Thread.sleep(10); } catch(Exception e){}
30         synchronized(C) {
31             System.out.println("038 acquired Lock C");
32             System.out.println("038 released all Locks\n");
33         }
34     }
35 }
36 }
37 }
38 class Task3 extends Thread []
39     Object A, B, C;
40     Task3(Object A, Object B, Object C) { this.A=A; this.B=B; this.C=C; }
41     public void run() {
42         synchronized(A) {
43             System.out.println("Thread-3 acquired Lock A");
44             try { Thread.sleep(10); } catch(Exception e){}
45         synchronized(B) {
46             System.out.println("Thread-3 acquired Lock B");
47             try { Thread.sleep(10); } catch(Exception e){}
48         synchronized(C) {
49             System.out.println("Thread-3 acquired Lock C");
50             System.out.println("Thread-3 released all Locks\n");
51         }
52     }
53 }
54 }
```

```
38 class Task3 extends Thread {
39     Object A, B, C;
40     Task3(Object A, Object B, Object C) { this.A=A; this.B=B; this.C=C; }
41     public void run() {
42         synchronized(A) {
43             System.out.println("Thread-3 acquired Lock A");
44             try { Thread.sleep(10); } catch(Exception e){}
45         synchronized(B) {
46             System.out.println("Thread-3 acquired Lock B");
47             try { Thread.sleep(10); } catch(Exception e){}
48         synchronized(C) {
49             System.out.println("Thread-3 acquired Lock C");
50             System.out.println("Thread-3 released all Locks\n");
51         }
52     }
53 }
54 }
55 }
56 public class Main {
57     public static void main(String[] args) {
58         Object A = new Object();
59         Object B = new Object();
60         Object C = new Object();
61         new Task1(A, B, C).start();
62         new Task2(A, B, C).start();
63         new Task3(A, B, C).start();
64     }
65 }
66 }
```

```
<terminated> Main [Java Application] C:\Users\Abdullah\
Abdullah acquired Lock A
Abdullah acquired Lock B
Abdullah acquired Lock C
Abdullah released all Locks

Thread-3 acquired Lock A
Thread-3 acquired Lock B
Thread-3 acquired Lock C
Thread-3 released all Locks

038 acquired Lock A
038 acquired Lock B
038 acquired Lock C
038 released all Locks
```