

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          }
10         synchronized(B) {
11             System.out.println("Abdullah acquired Lock B");
12             try { Thread.sleep(10); } catch(Exception e){}
13         }
14         synchronized(C) {
15             System.out.println("Abdullah acquired Lock C");
16             System.out.println("Abdullah released all Locks\n");
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         }
28         synchronized(B) {
29             System.out.println("038 acquired Lock B");
30             try { Thread.sleep(10); } catch(Exception e){}
31         }
32         synchronized(C) {
33             System.out.println("038 acquired Lock C");
34             System.out.println("038 released all Locks\n");
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         }
46         synchronized(B) {
47             System.out.println("Thread-3 acquired Lock B");
48             try { Thread.sleep(10); } catch(Exception e){}
49         }
50         synchronized(C) {
51             System.out.println("Thread-3 acquired Lock C");
52             System.out.println("Thread-3 released all Locks\n");
53         }
54     }
55 }

```

```
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         }
46         synchronized(B) {
47             System.out.println("Thread-3 acquired Lock B");
48             try { Thread.sleep(10); } catch(Exception e){}
49         }
50         synchronized(C) {
51             System.out.println("Thread-3 acquired Lock C");
52             System.out.println("Thread-3 released all Locks\n");
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
62         new Task1(A, B, C).start();
63         new Task2(A, B, C).start();
64         new Task3(A, B, C).start();
65     }
66 }
67
```

```
<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
```