## **Lab Program 10**

Q] Demonstrate Inter process Communication and deadlock

Lab Program 10

```
10 Demonstrate Inter Process Communication and
   deadlock
  class A 1
        Synchronized void fool B, b) 1
              Storing name = thread. aurent Thread().get No. 19 System. out. prontln (name+ "entired A. foo");
                Truad. Slup (1000);
             & catch CException e) {
                  System. out. prently ("A enterupted");
             System out. printhchamet + Frying to call B. laster").
             b.last();
       3
       Synchronised void last ()1
              sout (" Inside A. last ()");
class
         Sy rehronised
                          Void
                                 bar (A
               String name = Thread. reverent Thread (). get Name
               Sout (name + "entured B. bar")
                   Thread. Sley (1000);
              catch (Exception e) 5
                  sout (" B Porturupted')
              sout chame + " trying to call a last()");
              a. last L);
```

```
Eynchronised void last () 3
              sout (" Inside B. last ()");
class Peadlock implements Runnable ?
         A a = rew A();
         B b = hew B();
         Deadlock () 1
               Thread. current Thread(). Sot Name ("MainThread")
             Thread t= new Thread (this "Racing Thread"),
              t. start()
              a. foo os:
              Sout ("Backin main");
       public void run () 1
               b. bar (a);
              Sout (" Back i'n other thread ");
       4
        PSVM (Strig E) args) 1
                new Dead lock ();
  3
 output
 Main Thread entired A. foo
Main Thread trying to call B. Last ()
Racing Thread entired B. bar
Racing Thread troying to call B. Last ()
```

```
class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
```

Lab Program 10

```
try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
    synchronized void last() {
        System.out.println("Inside A.last()");
    }
}
class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B interrupted");
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
    synchronized void last() {
        System.out.println("Inside B.last()");
    }
}
class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b); // Main thread locks A and tries to call B.last()
        System.out.println("Back in main");
    }
    public void run() {
```

Lab Program 10 4

```
b.bar(a); // Racing thread locks B and tries to call A.last()
    System.out.println("Back in other thread");
}

public static void main(String[] args) {
    new Deadlock();
}
```

## //OUTPUT

```
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread entered B.bar
RacingThread trying to call A.last()
```

Lab Program 10 5