PROGRAM 10:

Demonstrate Inter process Communication and deadlock

1. Demonstration of Inter process Communication Observation

OBSERVATION:

```
10 persons trade Interprocess communication and decolocks
is Clama
  poolean valus et = false;
  synchronized and get ()
  1 while ( valued)
  fory & scritter -outgoingh ("soncines wasting");
      waster;
    b catch (Interrupted exception e)
     a system out possed in ("Got:" + in);
     Septemooul pred h ("got: "+h);
      valueset = fals;
      System out printin ("Internal Product");
      notify ();
       return n;
     Syndnowized void put(inth)
        while (value et)
              satur. out produl & proce waitings):
        addle (Intropted exception e)
        System. out provide ( other trupted exception caught ");
         y
terson = h;
         valusat = Poce;
        Santa road aparoth ( pad + h);
         · sutur -o at mut on [ " Turnail consumors)
         notfy();
```

clair Producer Emplements Runnable eq: Produces (9 eV). This or = q; new Thread (thi, "produser") start (); Public void sun() d int 8 = 09 while (icis) La put (it+); 3 clair consumer implements Romnouble Public state vaid main(shings ong) class pcfixed a border of the or how and po L gar=new (1); newpooduler(a); System out pritter(10 meen contrat (to 1 top4); Conquesting proof by confunts OUTEUT: (brulowally) ello (1. Impled enception 6) Soften out product of the superior plus aucht (1 1 bog) strage boo. it was Benotation of the fortunation of the

```
class Q { int n; boolean valueSet
int get() {
while(!valueSet) try {
System.out.println("\nConsumer waiting\n"); wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught"); }
System.out.println("Got: " + n); valueSet = false;
System.out.println("\nIntimate Producer\n"); notify(); return
} synchronized void put(int n) { while(valueSet)
System.out.println("\nProducer waiting\n"); wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
} this.n = n; valueSet
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n"); notify();
class Producer implements Runnable {
Producer(Q q) { this.q = q; new Thread(this,
"Producer").start();
} public void run() { int i =
0; while(i<15) {
q.put(i++);
```

```
class Consumer implements Runnable {
Consumer(Q q) { this.q = q; new Thread(this,
"Consumer").start(); }
public void run() { int
i=0; while(i<15) { int
r=q.get();
System.out.println("consumed:"+r); i++; }
} class PCFixed { public static void main(String
args[]) { Q q = new Q(); new Producer(q); new
Consumer(q);
System.out.println("Press Control-C to stop.");
```

OUTPUT:

Press Control-C to stop.
Put: 0
Intimate Consumer

Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer

Producer waiting
consumed:0
Got: 1
Intimate Producer
consumed:1
Put: 2
Intimate Consumer

Producer waiting
Got: 2
Intimate Producer
consumed:2
Put: 3

Intimate Consumer Producer waiting Got: 3 Intimate Producer consumed:3 Put: 4 Intimate Consumer Producer waiting Got: 4 Intimate Producer consumed:4 Put: 5 Intimate Consumer Producer waiting Got: 5 Intimate Producer consumed:5 Put: 6 Intimate Consumer Producer waiting Got: 6

Intimate Producer consumed:10 Put: 11 Intimate Consumer Producer waiting Got: 11 Intimate Producer consumed:11 Put: 12 Intimate Consumer Producer waiting Got: 12 Intimate Producer consumed:12 Put: 13 Intimate Consumer Producer waiting Got: 13 Intimate Producer consumed:13 Put: 14 Intimate Consumer

Intimate Producer

consumed:6 Put: 7

Intimate Consumer

Producer waiting

Got: 7

Intimate Producer

consumed:7 Put: 8

Intimate Consumer

Producer waiting

Got: 8

Intimate Producer

consumed:8

Put: 9

Intimate Consumer

Producer waiting

Got: 9

Intimate Producer

consumed:9 Put: 10

OBSERVATION:

```
@ Demonstration of
  clan o
 d synchomized void fool Bb)
   string name = Thread. Current thread() get name();
     Sylu coul proulls (name + ventered A foo)?
         Thread isleep (1000);
        Catch (Exceptione)
        a squado out-problement rate supplied i);
         3
        sylun-out. prills (name + " trying to call B. lant ()");
        b. laster;
        Synchronized noid lautes
       a stepter coul pr. Ah (" msicle A-last");
  class 13
   Syn chronized vold basin a)
    & Strong name = Threado Current Thread (). gut name ();
       Streng home = Thread.
        SOP (name + " entered B. bar);
       try thread. sleep (1000);
       catch (Execution e)
           Systemsoud-providin(" & Introupted");
          System out routin (name + "trying to call A. (aut () ");
           a larter;
           Syndmonized 208d Cartes
          as your outoposation "Invide A-last");
```

```
class Deadlock Emplements Rusnable
a
  Aa=new Ac);
  13 b = new 13(1);
 Dead lock
   of
    Thread . awent Thread () . Selvome ("NDEn Thread");
   Threadt = new Thread (this, "Paraling Thread");
    t-stort co;
    a. 100(b);
    SOP (" Block in malin thread ");
   3
  Public void runco
   a b. bar (a);
     SOPI" Back in other thread ");
   Publis states void moin (shing origit)
   new Dead lock or;
                         (a almod histor lower on 2
   System out postled name in hypog to coul a fout to
```

SOURCE CODE:

```
class A
{ synchronized void foo(B b)
  { String name = Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
                                                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 A.last"); }
class Deadlock implements Runnable
 A a = \text{new } A(); B b = \text{new } B();
  Thread.currentThread().setName("MainThread");
  Thread t = new Thread(this, "RacingThread");
   t.start(); a.foo(b); // get lock on a in this thread.
                                                      System.out.println("Back in main thread");
public void run() { b.bar(a); // get lock on b in other thread.
 System.out.println("Back in other thread");
public static void main(String args[]) { new Deadlock(); }
```

OUTPUT:

```
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
^C
C:\Users\satis\OneDrive\Documents\ooj_lab>
```