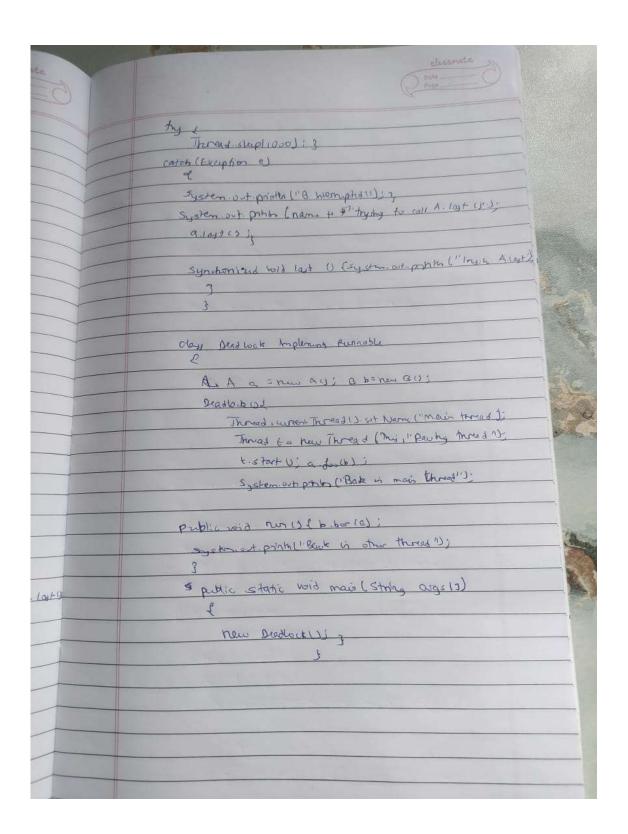
## Program 10

**Demonstrate Inter process Communication and deadlock** 

<b>E</b>			-
		Demonstrate later posen communication and disaster	-
		more and	_
		a vote lite exist	
11	1=)	Clay & L	
11		AL DI	4
1	-	bodien whise & John	
30			
		syndonized hit get of	
		1500 ( Valenser	
		ty e System out points ("consumer walther");	
12		System out porth ("Congruence	
		1 catch (interrupt of their	
		g (aught 11)	
		System at path ("Intereptates upton cought ");	
		system out printh ("Got!" In)	
		System : out. Path (" In Intimate Roducer (n'));	
		notify(1)	
		return,	
		3	
		Synchonized portional put (ht n)	
		Synthetize solveia per com	
		while lookerset)	-
1 4		try f	
1 4		system.out.prinths ("Producer waitingin")	
		ωα(+ () <sup>/</sup>	
		Jeath Interpreted Exception e) &	
1		System out path ("laterapted Exceptor caught"))	-
1		3	
		my n = n;	
		value set = trui;	
1		S. d. and the tree	
		System-out-path ("Put "+ h);	

	Quit C	
	Typtom out points ( Intimate Comune ")	
	not/s u;	
	1	
	The second second	
	A. miled	-
	class Ades cer implements Aumobbe &	-
	49	
	Poduur (O of)	
	new Thread (the "Producer") . start ();	
	3	
	Public pid run ut	
	int is of	
	while (ixis)	
	q. put (int)	
	3	
	clay consumer Emplement Runnable L	
	977	
	Royumur (9 8) (	
100000	1 9 9 1	-
	new Thread (this, "consume"). stort Us	
	1 11	
	public roid mus (It	
	(it is o)	
	while (icis) ?	
	M += q.get v;	_
	System out profiter ("Consumed!" Fr);	
	itt:	
	Jahred S. D. S. Line	
	North Company of the	
	Carried and a second and an arrange of the second	

	ELASSALLE Delte Page
	The state of the s
	clay Outling to main (5 thy warge) L
	1
	new Consume(g);  rew Consume(g);  System out posts ( Ky Contal - C to she ");
	3
17	Demonstration of doublish
	LUNG LOUIS I
	clays A
	Syndron 2rd wid fool B b)
	Stry new throad whethered 13-3d New (); System out proton brown + "entered A goo");
	thy the same of th
	Thread steep (1000); 3
	catch (Exception e)
	System out prints ("A "ntersuprit"); 3
	Systematiphin from + thing to call B. last (1) 16.18
	Systemated void last () (x
	System at prints ("Institute A log+")")
	Clay Be
	syndromed a good bar(A a) (
	String name = Thread (Uma +Th
	System out printer (name + "contend B. bar");



	al equips.	
& Jake Prous Comme	NO.	
Su fait		
OUNA	THE RESERVE OF THE PARTY OF THE	
Pot: 2 met		
to be longer		
P-D ducer	the district one	
get 0		
dutie		
Intrak product		
latinete Common	and district to	2
froduce waiting		
Conguented!	A Marie Mark	
got 1	A Land Committee of the Land Committee of th	
90	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Intimate Andrews	1 1 1 1 1	
Conjumed ! !	141.100 I	
14:2	0 ) = 0 11	
Internate Longismus	M. Laderradick and P.	
producer waiting	man A Company Assessment	S1 51
Got 12	an and then had a second	2
intrate poducer		
toyund 2	and the spirits 147	18
Put : 3	3	
1,41, 3		
	C. C	
Introde Consumer		
Producer waiting	et ill strategie	
Got: 3	Bullion B. B. B.	
1 atmate Poduce		
Conjuned: 3		
Pot : 4		
Intimate Consume		

	Deadlock program
	ou pul-
	mainThread entered A.foo
	RacingThread entered B. bear
	Rocing thread trying to call 1. last ()
-	Mainthread trying to call B. last ()
-	

```
//inter process communication
class Q {
  int n;
  boolean valueSet = false;
  synchronized 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 n;
  synchronized void put(int n) {
     while (valueSet)
       try {
          System.out.println("\nProducer waiting\n");
       } catch (InterruptedException e) {
          System.out.println("InterruptedException caught");
     this.n = n;
     valueSet = true;
     System.out.println("Put: " + n);
     System.out.println("\nIntimate Consumer\n");
     notify();
}
class Producer implements Runnable {
  Qq;
  Producer(Q q) {
```

this.q = q;

public void run() {
 int i = 0;
 while (i < 15) {
 q.put(i++);
}</pre>

}

Qq;

new Thread(this, "Producer").start();

class Consumer implements Runnable {

```
Consumer(Q q) {
     this.q = q;
     new Thread(this, "Consumer").start();
  public void run() {
     int i = 0;
     while (i \le 15) {
       int r = q.get();
       System.out.println("Consumed: " + r);
       i++;
}
class sync {
  public static void main(String args[]) {
     Q q = new Q();
     new Producer(q);
     new Consumer(q);
     System.out.println("Press Control-C to stop.");
}
// deadlock
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 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);
     System.out.println("Back in main thread");
  public void run() {
     b.bar(a);
     System.out.println("Back in other thread");
  }
  public static void main(String args[]) {
     new Deadlock();
```

```
C:\Users\shett\OneDrive\Documents\javaclasslab>javac sync.java
C:\Users\shett\OneDrive\Documents\javaclasslab>java sync
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
```

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: 6 Put: 7 Intimate Consumer Producer waiting Got: 7 Intimate Producer Consumed: 7 Put: 8 Intimate Consumer Producer waiting

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 Intimate Consumer Producer waiting Got: 10 Intimate Producer Consumed: 10 Put: 11 Intimate Consumer Producer waiting Got: 11 Intimate Producer Consumed: 11 Put: 12 Intimate Consumer

**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 Got: 14 Intimate Producer Consumed: 14 C:\Users\shett\OneDrive\Documents\javaclasslab> C:\Users\shett\OneDrive\Documents\javaclasslab>javac Deadlock.java

C:\Users\shett\OneDrive\Documents\javaclasslab>java Deadlock MainThread entered A.foo RacingThread entered B.bar RacingThread trying to call A.last() MainThread trying to call B.last()