```
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");
      wait();
    catch(InterruptedException e) {
      System.out.println("InterruptedException caught");
    this.n = n;
    valueSet = true;
    System.out.println("Put: " + n);
    System.out.println("\nIntimate Consumer\n");
```

```
synchronized void put(int n) {
  while(valueSet){
    try {
      System.out.println("\nProducer waiting\n");
      wait();
    }
    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 {
 Q q;
 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 {
  Q q;
  Consumer(Q q) {
    this.q = q;
    new Thread(this, "Consumer").start();
```

```
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 {
  Q q;
  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.");
}
```

D:\24BMSCE>javac PCFixed.java D:\24BMSCE>java PCFixed 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

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: 6

Put: 7

Intimate Consumer

Producer waiting

Got: 7

Intimate Producer

Consumed: 7

Put: 8







Intimate Producer Consumed: 9 Put: 10

Got: 9

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

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 D:\24BMSCE>









LAB - 10 Demonstrate Interprocess Communication a dead lock class Q 4 ent n; always boolean valueset = jaise; synchron: sed int get ()? Lystem. Out. paratio ("In. Consumor waitingla"); wast (); catch (Intersupted Exception e)? Jystem. Out. Parentla ("Intersupted Exception;
- caught"); System.out. prentin ("Got: "+n); Valueset = lalse; dystem. out. paientine "In Intimate producerla"; note(/ (); Hetounn; dynchron: 3ed void put (int n) 1 while (Value Set) & 19ml J.o.pin ("In. producer Wasting In"); 2 wast (); (atch (Intersupted Exception e)?
d.o.pln ("Intersupted Exception Caught");

95 [7] }

Main();

4.0.pln (" Interstopted Exception and 5

this. n = n;

value set = true;

\$.0.pln ("put: "+n);

	NO:	

J.o.pln ("In Internate Consumerine"); notify();	lass po Public Q q new U new U
	Public Rew New C
3 00	new C
	new C
3	newl
class producer implements Runnable?	
- 9; 32101 FE 13000 100 100 1	
Producer (O 0.)	3
new Theread (this, "peroducer"). Start();	26"
new Thread (this, "producer"). Start();	
The state of the s	/P:
Public void Huntil	14185
?n+ ?=0;	· put :
while (i'als)?	. (
8 PO+ (:++);	Intom
4 Cold of the cold	8 1000
3 consequence lus metaps	prod
Classification of account of	Got:
(lass Lonsomer emplements Runnable)	
(00 (10mm) (0 m))	Inter
Consumer (O g) ?	
thes. 9 = 9;	Produ
new Thread (this, "Consumer"). start();	Cons
Public void monch	Egot
Ont 9=0;	1100
whole (ic 15)?	
(110 0 10)	Consu
S. O. Pln ("consumed: 1"+91);	Got
14 + 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
3	Inter
3	0.1
1	produ
(0.17.109.) 119.08	401

class pe Fixed 1 Public static void main (staing augs[])?

Q e = new Q();

new producer(q);

new Consumer(q);

3.0.pln ("press Control - (to stap"); (":01 nables Start(); Press control - c to stop Put:0 Internate Consumer Producer Wasting Prot: 0 rabled Internate con producer Consumed: 0 start(); Got:1 Internated producer Consumed: Got: 2 Intermiate consumer Got: 2 maiting

```
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 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 = 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();
 }
}
```

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

Demonstration of Deadlock. 11a55 A 1 synchronised void joo (B b) (
string name = Thread, corrent Thread (). getName(); Lystem. out . Prentin (name + "entered A. (100"); facy ? Thread, 5/eep(1000); S. O. pln (" A Interrupted "); d. o. pln (name + " trying to call B. last 1)"); b. 195+(); synchron: 3ed void last () {
J.o.pin ("Inside A. last"); Class B1 synchronized void bay (A a) ? Jining name = Thread, consent Theread (). getName(); J.o.pin (name +" enteread B. ban "); fory? Thread, sleep (1000); S.o.pin ("B Interrupted");