PROGRAM 10:

Demonstrate Inter process Communication

10 A)

CODE:

```
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("\nNotify 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("\nNotify 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();
   public void run() {
       int i = 0;
       while (i < 15) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
    }
public 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:

```
Consumed: 1
Put: 2
Notify Consumer
Producer waiting
Got: 2
Notify Producer
Consumed: 2
Put: 3
Notify Consumer
Producer waiting
Got: 3
Notify Producer
Consumed: 3
Put: 4
Notify Consumer
Producer waiting
```

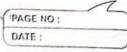
```
Got: 4
Notify Producer
Consumed: 4
Put: 5
Notify Consumer
Producer waiting
Got: 5
Notify Producer
Consumed: 5
Put: 6
Notify Consumer
Producer waiting
Got: 6
Notify Producer
Consumed: 6
Put: 7
Notify Consumer
Producer waiting
Got: 7
Notify Producer
```

```
Consumed: 7
Put: 8
Notify Consumer
Producer waiting
Got: 8
Notify Producer
Consumed: 8
Put: 9
Notify Consumer
Producer waiting
Got: 9
Notify Producer
Consumed: 9
Put: 10
Notify Consumer
Producer waiting
Got: 10
Notify Producer
Consumed: 10
Put: 11
```

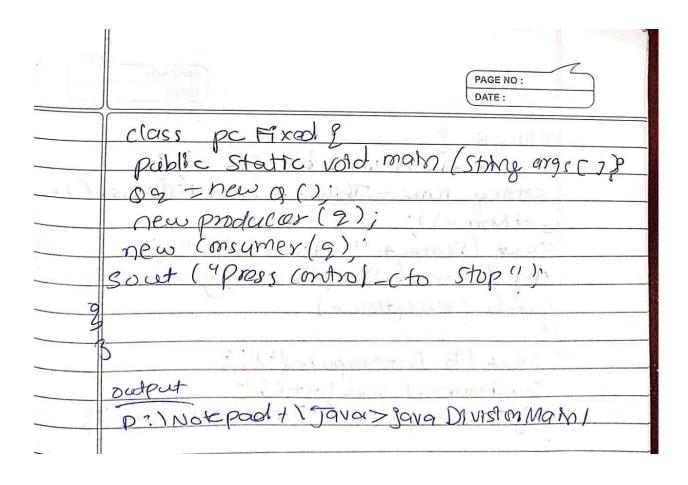
```
Notify Consumer
Producer waiting
Got: 12
Notify Producer
Consumed: 12
Put: 13
Notify Consumer
Producer waiting
Got: 13
Notify Producer
Consumed: 13
Put: 14
Notify Consumer
Got: 14
Notify Producer
Consumed: 14
```

NOTES:

	(DATE:
90	and deadlock.
	and deadlock.
-57	
	class of f
78 - 13	and n:
	boolean value set = false;
	Synchronizal Int get ()?
	coline (! valueset)
1907	System-out prottn (" In custimer wooding);
	walt ();
- Bu	2
,	catch (Introupted Exception e)?
	Soud (a Interrupted Exception e)?
11.	3
and my no	Sout ("Got "+n)
	valueset = false;
	Sout (" Intimate produces ");
	no+7 fy ();
	return n;
	3
	sychronized void put (Inta)s
	while walvo set)
	sow! Iproducer waifing");
	salf ()
	(goth (Intersupted Exceptione)
ry	Con (Trucky Control)
	I sout (" Intersupted Exteption agenti"),
	o soul file of the care of the
	Holan-p



73. 1 4	PAGE NO : DATE :
10.14	class produces implements Rumables
	producer (99) E
	new thread (+his " producor "). starti;
	Sill to the Clark of the half of the section of the
	int 1=0;
÷	10/0/2 (16/5)8
· (Fine	9. put (1++);
	9
70	2 - Land
	class consumer implements Runnables
	Og o
	Consumer (09)?
	new thread (this, u consumer ") . Starf();
	public void riencis
	while (1C15)5
	ant Y= 9. get().
	Sout (4 consumed (1+x);
	1 TENTHON TOHOUS AND A MAN I
	San to the same of
	6



10 B) DEMONSTRATION OF DEADLOCK

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();
}
```

```
System.out.println("Inside A.last");
synchronized void bar(A a) {
    String name = Thread.currentThread().getName();
   System.out.println(name + " entered B.bar");
        Thread.sleep(1000);
        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");
A a = new A();
B b = new B();
Deadlock() {
    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");
   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()
```

NOTES:

B	Demonstration of doodlock.
-	Charles Continued in the Continued in th
3	class A ?
	Synchromzad void fool Bb)
	2 - (78) 4 - 3 6
	Story rame - Thread Current Thread ().
	de et natue ().
	System out point in (name + "entered A-foo")
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	the PT and all all all all all all all all all al
500	try { Thread. Sleep(1000); } catch (Exception e)
	catch (tx(epflon e)
	& System. out partin (" A Interrupted")
	System out print in (name + trying to call
ar i	B. last ()4).
	b. (ast c) · 2 1) was been suches ?
	Synchronized vold (att () 2
	System.out partin(" Inside A. 1957 4); &
7)	Travelle about how hope I there I are st
	E graduater 153

PAGE NO: DATE : class B S Synchronized void box (A a){ Stang hame - Throad . corrent Thread (). get Nome V; Sout (name + " entered & bas"); try & Thread-Sleep(1000); & Catch (Exception e) Sout (B Interreputed"); & Synchrom zed void last () { Sy dem-out. Point in ("Inside A (ast); } class Deadlock Implementation Runable A a = new. A(); BB= new BC); Dead lock () & Thread - carrent Thread (). Set Name (4 Main Thread 11) Thread f = new Thread (this, "Racingt. Start (); a. foo(b); Sout ("Back in man thread"); public void nen (18 b. bor (a); Sout ("Back mother thread"); public static void many (string args())