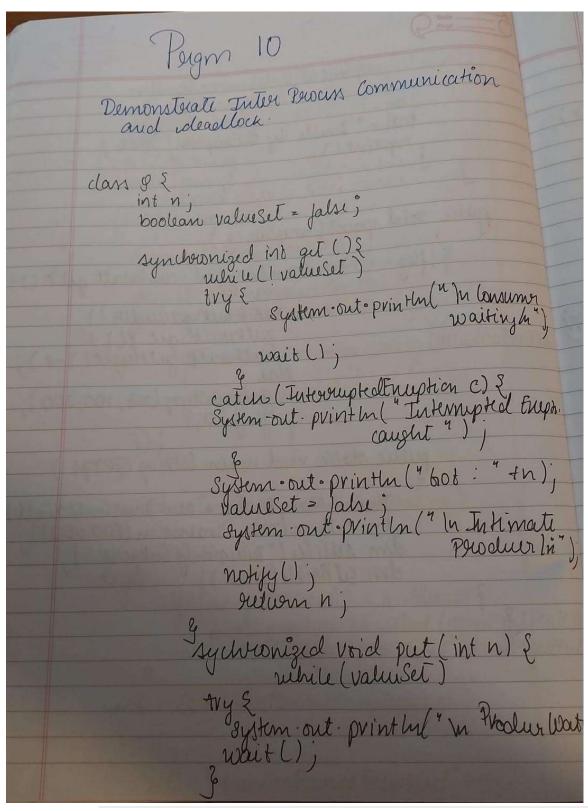
WEEK 10
Demonstrate Inter process Communication and deadlock



catch (Interoupted Enception e) {
System out printh ("Interrupted Enception eaught"); this n=n; valueSet = teme;
System. out. println(" Put: "+n):
System. out. println(" In Intimate Consumer by).

3 Mass Producer implements Runnable & Product (9 9) {
this 9, 29; new theread (this, " Peroduce "). start (); public void rum() {

int i = 0;

while (i < 15) {

q. put (i++); clars Consumer implements Runnable & consumer (99) & this 9 = 9; new theread (this, "Consumer") start (); public void run() &

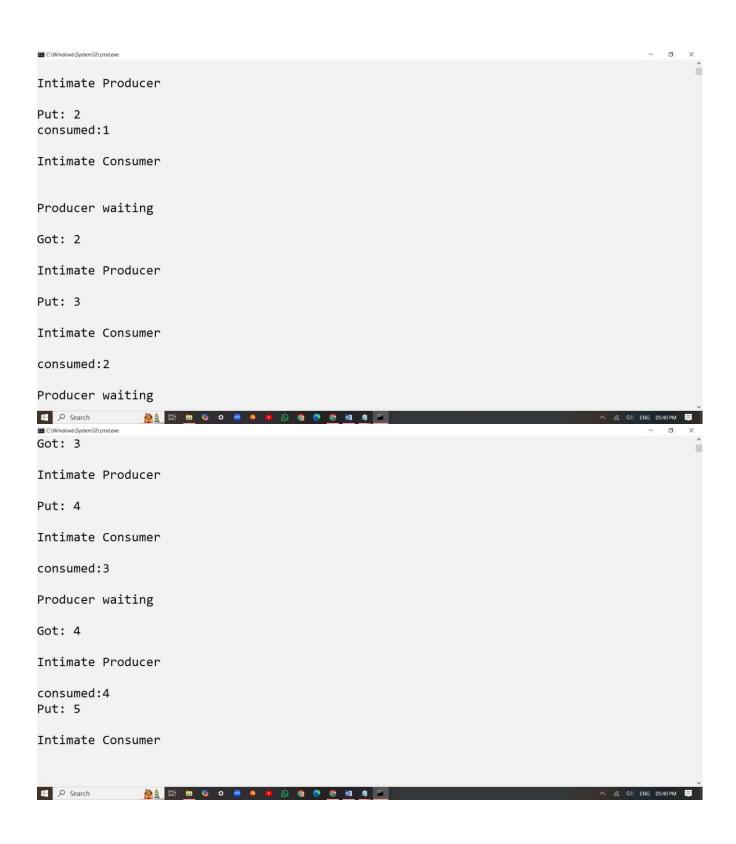
```
int i = 0;
while (i <15) &
      Mass PCFined
                                 usial main (String args
class Q {
```

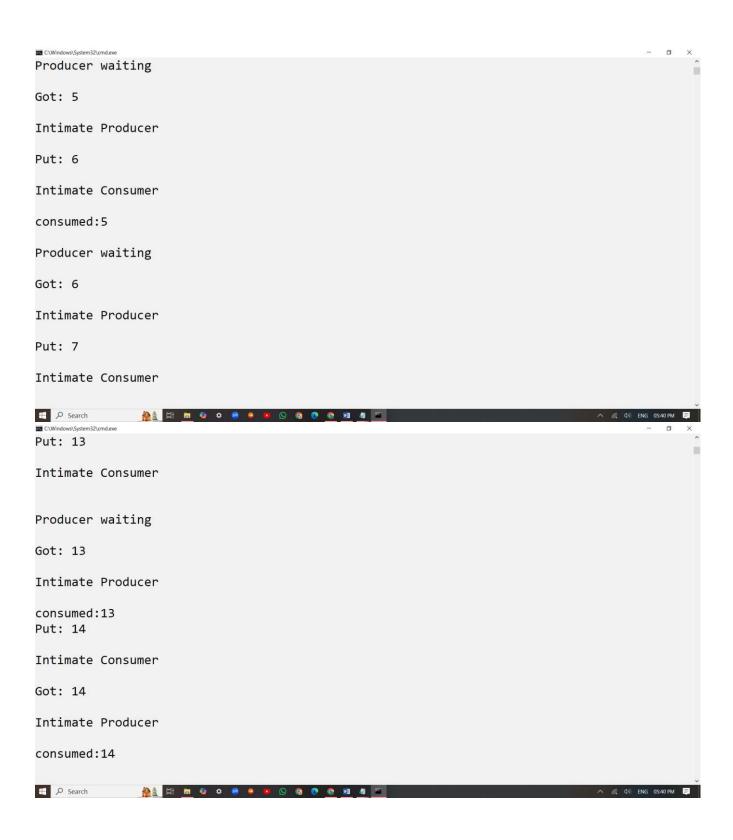
```
int n;
boolean valueSet = false;

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");
notify();
}
}
class Producer implements Runnable {
Qq;
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 {
Qq;
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.");
C:\Windows\System32\cmd.exe
                                                                                     - o ×
C:\Users\Srinivas\OneDrive\Desktop\BMS\SEM 3\PROGRAMS>javac PCFixed.java
                                                                                       Restore Down
C:\Users\Srinivas\OneDrive\Desktop\BMS\SEM 3\PROGRAMS>java PCFixed
Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
consumed:0
Got: 1
```





sclass A synchronized void foo(B b) gitNamel System out printh (name + "entered A foo"); Thread · Sleep (1000); latin (Enception e) of System out printhel A interrupted ", System out println (name + "trying to call Blast()"), b. last(); 3 synchronized void locust () & Sop.ln ("Turside A. last"); 3 class B & Synchronized void boor (A a) & String name = Thread · current Thread () System-out-println (name + "entered B. bar Thread slep(1000); 3 catch & Enupline) & System out print hel B Interrupted"; 3.0.7 In [ name t "trying to call A last[]] synchronized void last 1) of 8.0. Pln 1" Turide A-last

```
class Deadlock implements Runnable
      Aa = new A ();
Bb = new B();
Deadlock () {

Thread : wirent Thread () : sit Namel "Main".

Thread t = new Thread (this, "Racing Thread")

t : Start ();
                a. foo (b);
System. out. print m("Back in main thread)
System out print In (" Back in other thread

public static void main (String (Jargs) {

new Deadlock ();
                 bobarla);
```

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