

6/2/24

LAB-10

Demonstrate inter communication and deadlock

class Q {

int n;

boolean valueset = false;

synchronized int get() {

while (!valueset)

try {

System.out.println("Consumer waiting\n");

wait();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");

}

System.out.println("Get: " + n);

valueset = false;

System.out.println("\nIntimate producer\n");

notify();

return n;

}

synchronized void put(int n) {

while (valueset)

try {

System.out.println("In Producer waiting\n");

wait();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");

}

this.n = n;

valueset = true;

System.out.println("Put: " + n);

System.out.println("Intimate Consume\n");

notify();

classmate

Date
Page

classmate

Date
Page

}

{

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 consume 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 v = q.get();

System.out.println("Consumer: " + i);

i++;

}

}

}


```
class Pc fixed {
```

```
public static void main (String args []) {
```

```
    Gc = new Gc();
```

```
    new producer (q);
```

```
    new consume (q);
```

```
    System.out.println ("Press control -c to stop");
```

Output:-

put 0

intimate consume

Producer waiting

got 0

Intimate Produces

consumed: 0

consume waiting

Put: 1

Intimate consumer

Producer Waiting

Got: 1

Intimate produce

consumed: 1

Put: 2

Intimate Consume

Producer waiting

Got: 2

Put: 3

Intimate consumer

Producer waiting

Got: 3

Intimate consumer

consumed: 3

Put: 4

Shreyas.K

(BM2255271)