

Demonstrate Inter process Exercise

```
class Q
{
```

```
    int n;
    boolean valueSet = false;
```

```
    synchronized int get() {
        while (!valueSet)
            try {
```

```
                System.out.println("In consumer waiting\n");
                wait();
```

```
            } catch (InterruptedException e)
            {
```

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

```
        System.out.println("Got: " + n);
```

```
        valueSet = false;
```

```
        System.out.println("In Intimate 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("\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();

}

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.");
```

```
}
}
```

Output:

Press Control-C to stop.

Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate ~~Consumer~~ Producer

Put: 1

Intimate Consumer

Consumed: 0

Producer waiting

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

Producer waiting

Got: 4

Intimate Producer

Consumed: 4

Put: 5

Intimate Consumer

Producer waiting

Got: 5

Intimate Producer

Consumed: 5

Put: 6

;

Shreyas Rao M

IBM22CS272