

13/08/24

LAB =

a) * Demonstrate inter process communication
and deadlock:

```
class Q {  
    int n;  
    boolean valueSet = false;  
    synchronized int get () {  
        while (!valueSet)  
            try {  
                System.out.println ("Consumer waiting");  
                wait ();  
            } catch (InterruptedException e) {  
                System.out.println ("InterruptedException  
caught");  
            }  
            System.out.println ("got : " + n);  
            valueSet = false;  
            System.out.println ("Intimate producer");  
            notify ();  
            return n;  
        }  
        synchronized void put (int n) {  
            while (valueSet)  
                try {  
                    System.out.println ("Producer waiting");  
                    wait ();  
                } catch (InterruptedException e) {  
                    System.out.println ("InterruptedException  
caught");  
                }  
            this.n = n;  
            valueSet = true;  
        }  
    }  
}
```

```

        System.out.println("Put:" + n);
        System.out.println("In Intimate Consumer  
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 peFixed {
    public static void main (String args []) {
        A q = new A ();
        new Producer (q);
        new Consumer (q);
        System.out.println ("Press Control-C to stop.");
    }
}

```

output

put : 0

Intimate Consumer

Producer writing

Press control-c to stop

Got: 0

Intimate producer

Consumed: 0

Put: 1

Intimate Consumer

producer writing

Got: 1

Intimate producer

Consumed: 1

put: 2

Intimate Consumer

producer writing

Got: 2

Intimate producer

Consumed: 2

put: 3

Intimate Consumer

producer writing

Plot: 3

Intimate Producer

Consumed: 3

Plot: 4

Intimate Consumer

Producers waiting

Plot: 4

Intimate Producer

Consumed: 4

Plot: 5

Intimate Consumer

Plot: 5

Intimate Producer

Consumed: 5

Shashank SP

BM@CS256