

```
class BMSThread extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println ("BMS College  
of Engineering");  
            try {  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
class CSEThread extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println ("CSE");  
            try {  
                Thread.sleep(2000);  
                // sleep for 2 sec  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

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

// to create & start first thread
BMSThread bms_Thread = new BMSThread();
bmsThread.start();

```
// Create & start the second thread  
CSEThread cseThread = new CSEThread();  
}
```

Output: `(nx3 int) x3tub(1) nida)`

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

B.M

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

Bmc

~~BMS College of Engineering~~

LAB Program 10.

Date / /
Page / /

ms Thread
(),
cad
d();

```
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 = true;  
}
```

```
System.out.println("In Producer Waiting(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;
    value.set=true;
    System.out.println("Put: "+n);
    System.out.println("In Intimate Consumer(n);");
    notify();
}
}

```

```

class Producer implements Runnable
{

```

```

    Queue q;
    producer(Q q)
    {

```

```

        this.q=q;
        new Thread(this, "Producer").start();
    }
    public void run()
    {

```

```

        int i=0
        while(i<5)
        {
            q.put(i++);
        }
    }

```

```

class consumer implements Runnable
{

```

```

    Queue q;
    consumer(Q q)
    {

```

```

        this.q=q;
        new Thread(this, "Consumer").start();
    }

```

```

    public void run()
    {

```

```
int i=0;  
while(i<5)  
{  
    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:

Put : 0

Intimate Consumer

Producer waiting

Cout : 0

Intimate Producer

Put : 1.

Intimate Consumer

Producer waiting

consumed : 1

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 Consumers

Date _____
Page _____

got : 4

Intimate Producer

consumed : 4

~~Intimate
Producer~~