

#WEEK-8: (Thread Program)

class BMSThread extends Thread

{

 public void run()

{

 while (true)

{

 System.out.println("BMS College Of Engineering");

 try

{

 Thread.sleep(10000);

}

 catch (InterruptedException e)

{

 e.printStackTrace();

 }

}

}

class CSEThread extends Thread

{

 public void run()

{

 while (true)

{

 System.out.println("CSE");

 try

{

 Thread.sleep(2000);

}

 catch (InterruptedException e)

{

 e.printStackTrace();

}

}

}

```
public class ThreadExample
{
    public static void main (String [] args)
    {
        BMSThread bmsThread = new BMSThread ();
        bmsThread.start ();
        CSEThread cseThread = new CSEThread ();
        cseThread.start ();
    }
}
```

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

Dee

LAB - PROGRAM 10:

TPC:

```
class A {
    int n;
    boolean valueset = false;
    synchronized int get() {
        while (!valueset)
            try {
                System.out.println("In consumer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Got: " + n);
        valueset = false;
        System.out.println("Notify 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");
            }
        valueset = true;
    }
}
```

```
catch (InterruptedException e) {  
    System.out.println("InterruptedException caught");  
}  
  
this.n = n;  
valueSet = true;  
System.out.println("Put: " + n);  
System.out.println("In Intermediate Consumer");  
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

{

 Q 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 PC fixed {

 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 Producer

Put: 1

Intimate Consumer

Producer waiting

Consumed: 0

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

Got: 4

Intimate Producers

consumed: 4

~~C&L
City
10/12/11~~