6/2/24

Lab Program 8

MultiThreading

```
class bms extends Thread{
    public void run(){
        for(int i=1; i<=5; i++){
            System.out.println("bms college of engineering");
            try
            {
                Thread.sleep(10000);
            }
            catch(InterruptedException e){
                e.printStackTrace();
            }
        }
    }
class cse extends Thread{
    public void run(){
        for(int i=1; i<=10; i++){
            System.out.println("cse"+i);
            try{
                Thread.sleep(2000);
            }
            catch(InterruptedException e){
                e.printStackTrace();
            }
        }
    }
class threadMain{
    public static void main(String a[]){
        bms obj1 = new bms();
        cse obj2 = new cse();
```

obj1.start();
obj2.start();
}
}

Output
bms college of engineering1
cse 1
cse 2
cse 3
cse 4
cse 5
bms college of engineering 2
cse 6
cse 7
cse 8
cse 9
cse 10
bms college of engineering 3
bms college of engineering 4
bms college of engineering 5.

6/2/24 Lab Program 10

Interprocess communication

```
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("Interrupted Exception
                                caught");

        }
    }
    System.out.println("Got:" +n);
    valueSet = false;
    System.out.println("Intimate Producer\n");
    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("Intimate 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<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 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.");
    }
}
```

O/P Put: 0
Intimate Consumer

Producer waiting
Press Control-C to stop.
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 consumer

Producer waiting
Put: 4
Intimate consumer

Got: 4
Intimate consumer

consumed: 4

6/2/24