lab-10-A

Demonstrate Inter Process communication & deadlock → class Q {
int n
Synchronized int get() {
System.out.println("Got:" + n);
return; }

Synchronized void put(int n) {
this.n = n;
System.out.println("put:" + n);
}}

class procedure implements Runnable {
Q p;
Procedure (Q p) {
this q = q;
new Thread (this, "procedur").start()
}
Public void run() {
int i = 0;
while (i < 8) {
q = put(i++);
}}}
class consumer implements Runnable {
char q;
Consumer (Q q) {
this.q = q;
new Thread (this, "consumer").
start();
}.

```
Public void run(){
    int i=0;
    while(i<8){
        int r=q.get();
        i++;
}}}

Class pc{
public static void main(String args[])
{   Q q = new Q();
    new procedure(q);
    new Consumer(q);
    System.out.println("Press control-c
    to stop");}}
```

Output:

```
put 0
Got 0
put 1
Got 1
Put 2
Got 2
Put 3
Got 3
Put 4
Got 4
Put 5
Got 5
```

Lab-10-B

```
class A {
    Synchronized void foo(B b){
        String name = Thread.currentThread().get
        Name();
        System.out.println(name+"Entered A.foo");
        try {
            Thread.sleep(1000);
        }
        catch (Exception e){
            System.out.println("A Interrupted");
        }
        System.out.println(name+"trying to call
        B.last()");
        b.last();
    }
    void last(){
        System.out.println("Inside A.last");
    }
}

class B {
    Synchronized void bar(A a){
        String name=Thread.currentThread().
        getName();
        System.out.println(name+"Entered B.
        bar");
        try {
            Thread.sleep(1000);
        }
        catch(Exception e){
            System.out.println("B Interrupted");
```

```
        System.out.println(name+"trying to call
            A.last()");
        a.last();
    }
    void last(){
        System.out.println("Inside A.last");
    }
}

public class Deadlock implements Runnable{
    A a = new A();
    B b = new B();
    Deadlock(){
        Thread.currentThread().setName
        ("MainThread"); Thread t = new Thread
    t.start();                    (this, racing in other thread)
    a.foo(b);
    System.out.println("Back in main
    thread");
}
public void run(){
    b.bar(a);
    System.out.println("Back in other
    thread");
}
Public static void main(String args[])
{    new Deadlock();
}
}
```

output:

Racing in other thread entered B.bar.
Main thread Entered B.foo
Racing in other thread trying to call.
A.last()
Inside A.last
Back in the other thread.
Main thread trying to call B.last()
Inside A.last
Back in Main thread.

13.02.24