Date:
YOUVA

Lab 10 : Demonstrate Inter process communication and deadlock.

```
class Q {
    int n; boolean valueset = false;
    synchronised int get() {
        while (!valueset) {
            try {
                system.out.println("\n Consumer
                    Waiting \n");
                wait();
            } catch (Interrupted Exception e) {
                System.out.println("Interrupted Exception
                    caught");
            }
        }
        System.out.println("Got :" + n);
        valueset = false;
        system.out.println("\n Intimate Producer\n");
        notify();
        return n;
    }

    synchronised void put (int n) {
        while (valueset) {
            try {
                System.out.println("\n Producer
                Waiting \n");
                wait();
            } catch (Interrupted Exception e ~~caught~~") {
                System.out.println("Interrupted Exception
                    caught");
            }
        }
        this.n = n;
```

```
            valueset = true;
            System.out.println ("Put :" +n);
            System.out.println ("\n 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();
            i++;
        }
    }
}
```

M T W T F S S

Page No :

Date

YOUVA

```
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
Got : 0
Put : 1
Got : 1
Put : 2
Got : 2
```

[illegible]

13-2-24

Deadlock :

```
class A {
    synchronised void foo (B b) {
        String name = thread. currentThread (). getName()
        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 {
    Synchronised 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");
    }
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock () {
        Thread.currentThread().setName ("Main
                Thread");
        Thread t = new Thread( this, "Racing Thread"
        t.start ();
        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:

```
Main Thread entered A.foo
Racing Thread entered B.bar
Racing Thread trying to call A.last()
Inside A.last
Main Thread trying to call B.last()
Inside A.last()
Back in main thread
Back in other thread
```

[illegible] 13-2-24