

Lab 10

classmate

Date _____

Page _____

(10)(a)

```
→ class Q {
    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("Interrupted Exception caught");
            }
        }
        System.out.println("Got :" + n);
        valueSet = false;
        System.out.println("Notify Producer\n");
        notify();
        return n;
    }

    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("Producer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        }
    }
}
```

```
this.n = n;  
valueset = true;  
System.out.println("Put:" + n);  
System.out.println("In Notify 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 < 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();  
        System.out.println("Consumed  
        " + r);  
        i++;  
    }  
}
```

```
public 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: 1

Got: 1

Put: 2

Got: 2

(10) (b)

classmate

Date _____

Page _____

→ public class DeadlockExample {

public static void main (String [] args) {
Object lock1 = new Object ();

Object lock2 = new Object (),

Thread thread1 = new Thread (() → {
synchronized (lock1) {
System.out.println ("Thread
1 acquired lock1");

try {

} Thread.sleep (100);
} catch (InterruptedException e) {
e.printStackTrace ();
}

synchronized (lock2) {

System.out.println ("Thread 1
acquired lock2");

}

}

},

Thread thread2 = new Thread (() → {
synchronized (lock2) {
System.out.println ("Thread 2
acquired lock2");

try {

} Thread.sleep (100);

} catch (InterruptedException e) {
e.printStackTrace (); }

synchronized (lock1) {

System.out.println("Thread

2 acquired lock2");

3;

thread1.start();

thread2.start();

3;

4;

12;

13;

Output:

Thread 1 acquired lock1.

Thread 2 acquired lock2.