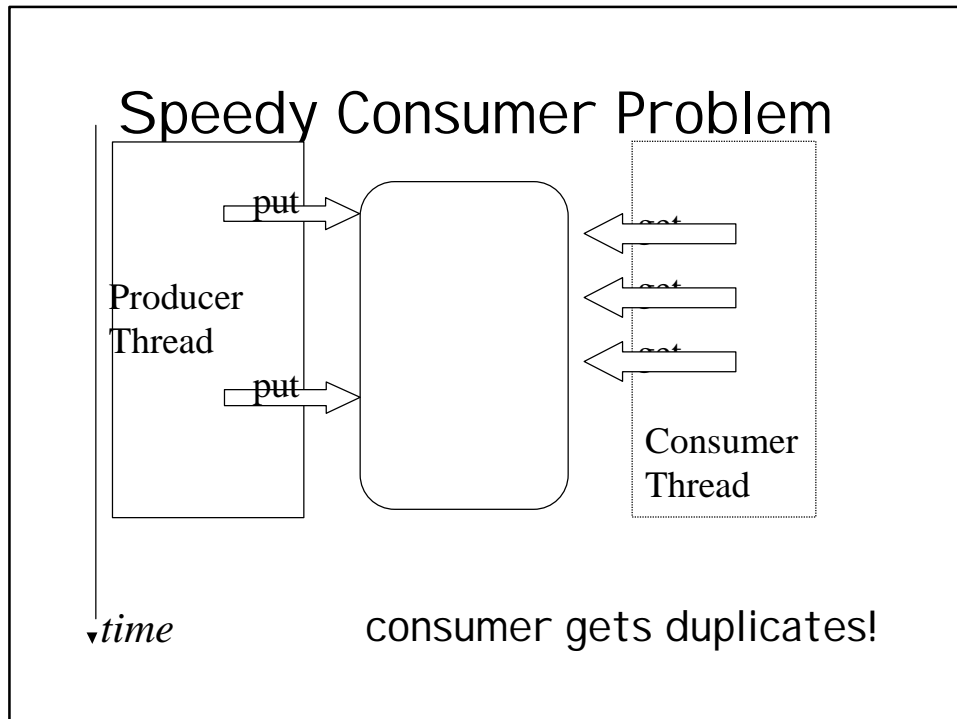
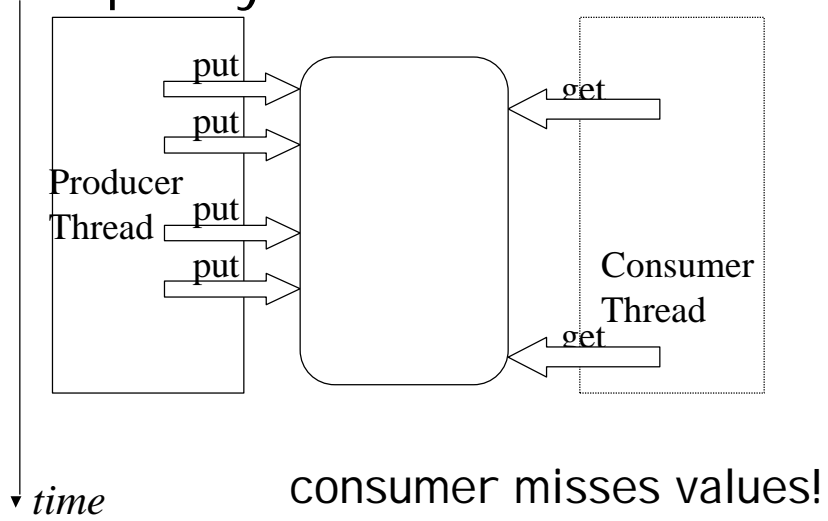


Thread Coordination with Notify and Wait

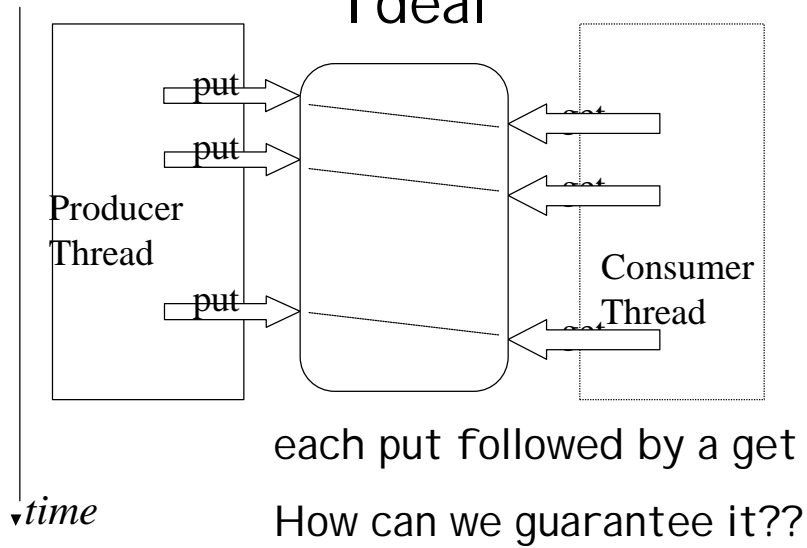
(Part II)



Speedy Producer Problem



I deal

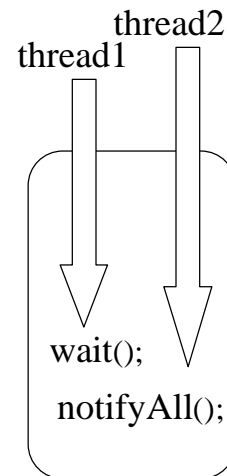


notify and wait - the difficulty

if thread2 goes first,
thread1 will miss what
thread2 did

thread1 may wait forever

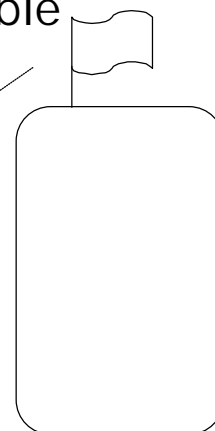
needed: notify & wait & a
flag called available



notify and wait

available

**we set this flag when a
producer has deposited
something for a consumer to
get**



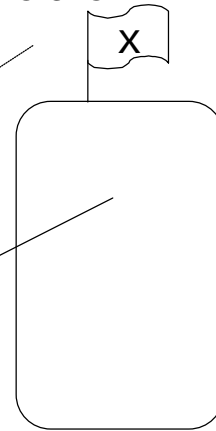
cubbyhole

notify and wait not available

indicates the cubbyhole object has no value -- nothing to get

initial state

`boolean available = false;`



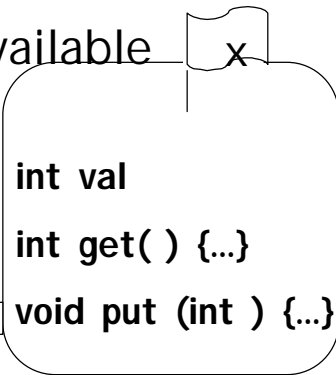
cubbyhole

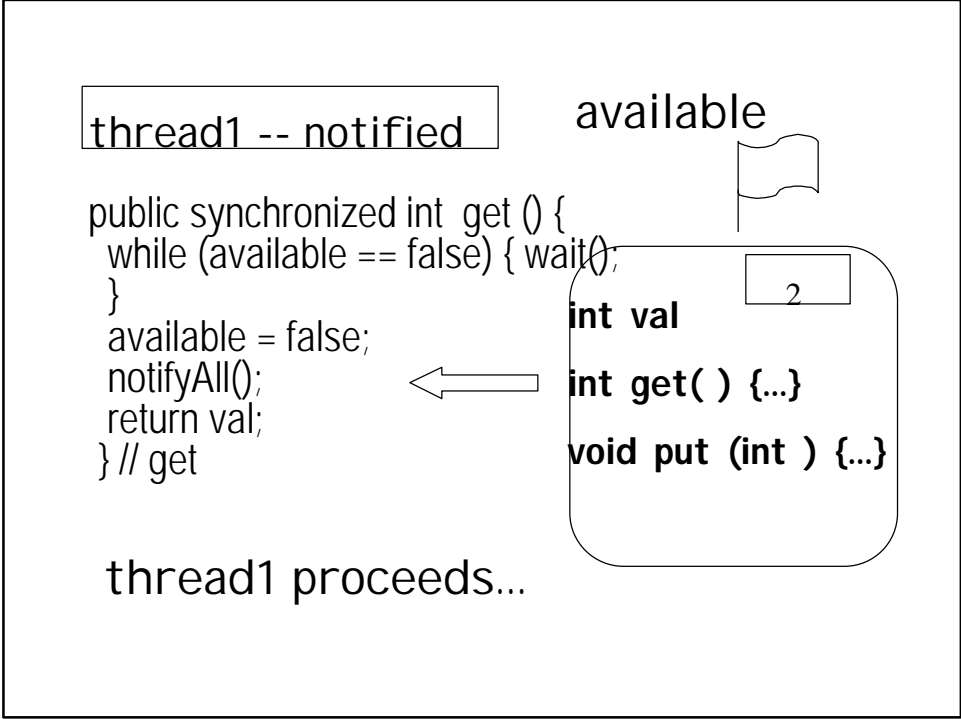
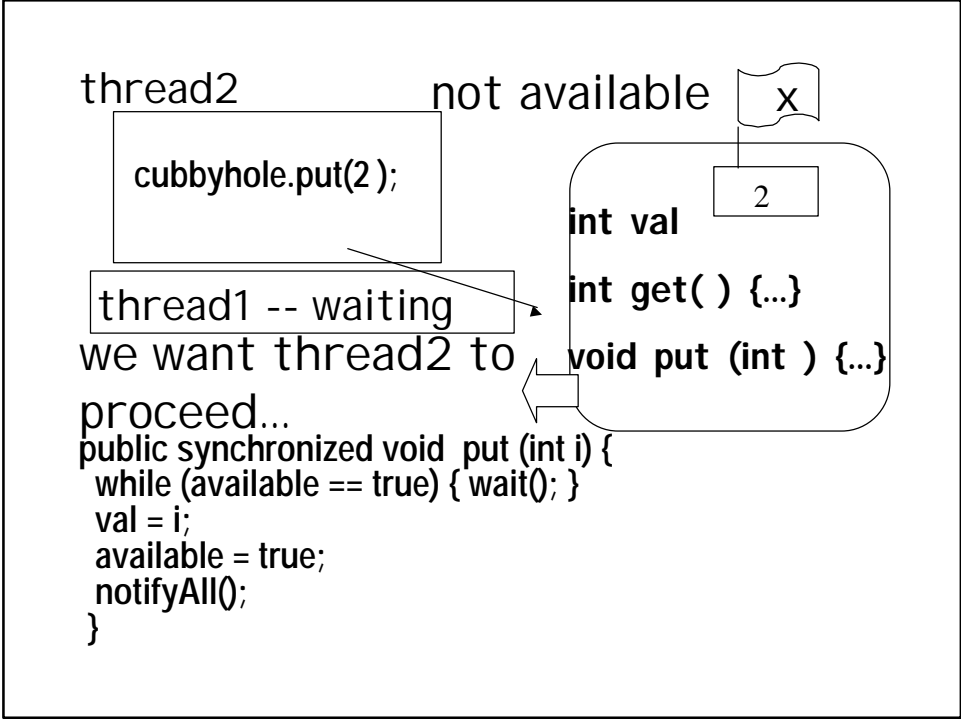
thread1 not available

`k = cubbyhole.get();`

we want thread1 to wait until

```
public synchronized int get () {  
    while (available == false) { wait(); }  
    available = false; // because I got it!  
    notifyAll(); // other deliveries may be waiting  
    return val;  
} // get
```





```

public synchronized int get () {
    while (available == false) {
        try { wait(); }
        catch (InterruptedException e) { }
    }
    available = false;
    notifyAll();
    return val;
} // get

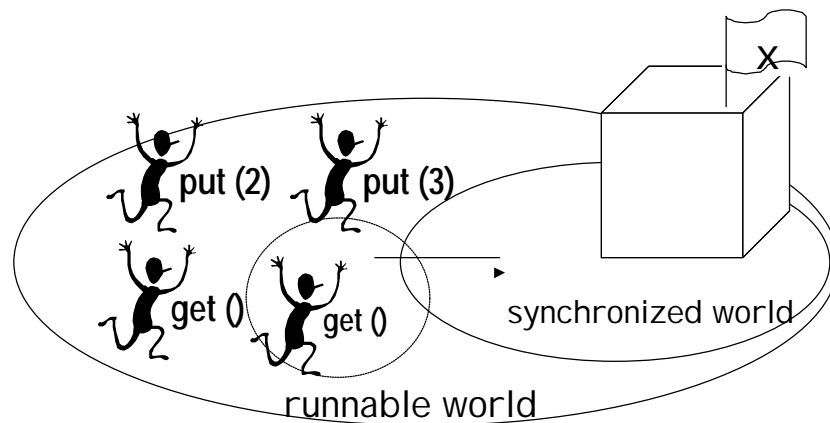
```

The Complete Code!

```

public synchronized void put (int i) {
    while (available == true) {
        try { wait(); }
        catch (InterruptedException e) { }
    }
    val = i;
    available = true;
    notifyAll();
} // put

```



One runnable thread gets picked to execute

