Threads

How are the steps of the threads interleaved?

- Thread scheduler
  - A: A B F G
  - B: J K L F G H I C H I M N D E
Thoughts on concurrency control:

- Let A complete before B starts
- Let them run according to the thread scheduler
- If method 3 is "critical" (don't want both in there at the same time) place a lock around method 3
- Too much control hurts performance.

how to manage interleaving of concurrent threads
Java thread support — creating & starting threads

- Thread class
  
  `run()` — what the thread does —

  override this

  `sleep(int milliseconds)` — static

  throws `InterruptedException`

  `join()` — returns when this thread terminates

- `start()` — starts the thread

  (in turn, `start` calls `run` within the thread)

- `Runnable` interface — public `void run()`

  `(new Thread(myRunnableObject)).start();`
Concurrency control options

- join
- locking

```java
ReentrantLock
myLock.lock();
// do critical stuff
myLock.unlock();
```

- synchronized methods

```java
public synchronized void foo() {
    notify();
}
```

```
x.foo(); // get a lock on x on entry +
release lock on exit
notify(); // notify waiting thread that releasing lock
x.wait();
```
Event Queue compression

| 3 | 10 | 8 | 6 | ... | 2 | 9 |

mouse Draggable