ECE 462
2/11/3/2010

GREEN = JAVA

```
class Data {
 public:
  Data (int a, int b) { x=a; y=b; }
  void f() {
    int r = rand()%100;
    x += r;
    y -= r;
  }
  int sum () {
    int s = x+y;
    return s;
  }
 private:
  int x;
  int y;  QMutex mu;
};
```

(2) DATA ONLY — synchronized public; mu.lock() → … mu.unlock()

synchronized public int sum () { mu.lock() → … mu.unlock() → }

Math.random()

```
class MyThread : public QThread {
 public:
  MyThread (Data * a) { d = a; }
  void run () {
    for (int i = 0; i < 100; i++) {
      d->f();
    cout << d->sum() << endl;
    }
    System.out.println(d.sum());
  Data * d;
};
```

(3) THREAD ONLY

```
Data * d = new Data (50, 50);
MyThread * t1 = new MyThread (d);
        * t2
        * t3
```

(1) DATA + THREAD

t1->start(); t2 ... t3 ...
t1->wait(); t2 ... t3 ...

```
class Account {
 private:
  QMutex mu;
  QWaitCondition cond;
  int balance;
 public:
  Account () { balance = 100; }
  void deposit (int d) {
    mu.lock();
    balance += d;
    cond.wakeAll(); // cond.notifyAll();
    mu.unlock();
  }
  void withdraw (int w) {
    mu.lock();
    while (balance < w) {
      cond.wait (&mu);
    }
    balance -= w;
    mu.unlock();
  }
  void print () {
    mu.lock();
    cout << balance << endl;
    mu.unlock();
  }; // class Account
```

synchronized — busy waiting

// temporarily release lock & inform when condition changes

```
try {
  wait();
} catch (InterruptedException e) { ... }
```

new *

new *