Threads

In Java, the Thread class & Runnable Interface

Create a class that implements Runnable; in our program we create a thread & give it our Runnable.
```java
public void static main () {
    Thread t1 = new Thread(new R1());
    Thread t2 = new Thread(new R1());
    Thread tn = new Thread(new R1());

    t1.start();
    t2.start();
    tn.start();
}
```

This code is a thread 'main' thread.
All programs have a main thread.

All GUI programs have at least 2 threads
1. main thread
2. graphics event thread
"Z" problems w/multithreading

1. possibility for race conditions
   possibility of deadlock
2. in determinate nature
   makes testing & debugging difficult
deadlock

see notes about Banking problem

problem is

2 threads simultaneously manipulating a bank balance.

Solution: allow threads to gain exclusive access to a resource.
Matthew <i>xfers</i> $50 to Bell. Bell <i>xfers</i> $50 to Matthew

1. Need exclusive access to Bell's balance.
2. Needs exclusive access to Matthew's balance

* We can't set it yet.
2. Needs exclusive access to Bell's account but can't get it

⇒ goes to sleep

Dearロック！
Advantages to Multithreading

1. More efficient use of resources

2. Can make our program easier.

Erlang

Ericsson's Programming Language
Event-driven simulations

People arrive randomly at bank

They randomly choose what queue to get in.
Multithreaded Aprch

Create a thread

Person

Arrive at bank
Do transaction
Quiz

1. What is a (Java) interface?
2. What is a (Java) thread?