

Synchronization Notes

1. Review race condition
 - a. `new Thread()`
 - b. `Runnable.run`
 - c. `Thread.start()`
 - d. Who gets to write to `balance` first?
2. Coarse control over what happens before/after:
 - a. `Start()`
 - b. `Join()`
3. Atomic classes (`java.util.concurrent.atomic`)
4. Synchronize statement
 - a. Example: Synchronizing access to `ArrayList`
5. Synchronized methods
 - a. Example: making our own synchronized `Array`.
6. Three design alternatives

Next time: When good synchronization goes bad...