CS 342 Software Design

- **Complete the course evals**
- **Last HW out soon**
- **Today:**
  - **Concurrency and Synchronization**

Concurrency: when multiple sequences of operations are run in overlapping periods of time.

- **Physically concurrent = at the same time**
- **Logically concurrent = sometimes time-sharing on a single CPU**

Race Conditions: When two or more threads attempt to update mutable shared data at the same time.

Mutable: you can change the states and fields after the object is created.

Synchronization: Preventing thread interference and memory consistency errors

The Java programing language provides two basic types of synchronization:

- ***synchronized methods***
- ***synchronized statements***

How it works:

Every object has an intrinsic lock associated with it.

A thread that wants exclusive and consistent access to an object's fields must acquire that objects lock and then release the lock when finished.

When a thread releases the lock a “happens-before” relationship is established between that action and any subsequent acquisition of the same intrinsic lock

Synchronized methods: add keyword
public synchronized foo1(){}

- Only one thread can invoke that method at a time for the same object. All other threads suspend execution till the first thread is done.
- When the synchronized method exits, it creates a “happens-before” relationship. Guarantees that changes to the state of the object are visible to all threads.

Synchronized Blocks:

synchronized(this){ //do something }

- Unlike synchronized methods, blocks must specify the object providing the lock.
- You can create objects whose only purpose is to supply locks

Problem: DeadLock: https://www.toptal.com/software/introduction-to-concurrent-programming

Two or more threads are blocked forever, waiting on each other.