Java Threads

Thread and Runnable
start(), join() and sleep()
Races and synchronized
Thread and Runnable

• The `Thread` class is used to create threads and interact with them.

• Two ways to create a thread:
  1. Subclass `Thread`, extending its `run()` method.
     • Advantages: class inherits all of `Thread`’s methods
     • Disadvantages: can’t subclass anything else
  2. Use the `Runnable` interface and implement its `run()` method.
     • General, but does not inherit `Thread`’s methods
start(), join() and sleep()

- Calling `t.start()` will start execution of the `run()` method within the thread `t` (and continue with execution of the current thread).
- Calling `t.join()` will cause the current thread to wait until thread `t` terminates.
- Calling `Thread.sleep(ms)` will cause the current thread to go to sleep for `ms` milliseconds.
Races and the \texttt{synchronized} keyword

• Too many cooks…
  – Coordination is the big challenge of concurrency
  – How do we avoid conflicts?
  – How do we impose some level of coherence and order?

• A ‘race condition’ is a situation where one or more threads race non-deterministically to be the first to read or write a variable

• The \texttt{synchronized} keyword
  – Qualify a method, ensures only one thread executes that method at any time