Delegation - Atomics

Delegation

- It didn't take long of using threads before we ran into issues with **Shared Mutable State**
- One way to handle this is to let someone else do the work
- This is known as **Delegation**
  - In delegation you use pre-existing classes that are already thread safe
- **Thread Safe** is a term used to describe classes and operations that behave correctly in a multithreaded environment

Atomic Types

- Java has many delegate classes in it
- The first one we are going to discuss is atomic types
- They are in the `java.util.concurrent.atomic` package which was introduced in Java 5
- The types that have supported atomic versions are:
  - AtomicBoolean
  - AtomicInteger
  - AtomicIntegerArray
  - AtomicLong
  - AtomicLongArray
  - AtomicReference
- Atomic referes to an operation that cannot be subdivided
  - In Java, that means effectively nothing else will happen in the middle of performing an atomic operation

Activity: Counter (10000x100)

- Modify the Counter activity you created previously to use an AtomicInteger instead of int
  - You can find the available methods in Oracle's Java Documentation
- What happens when you run the application after making any necessary changes?

DISCUSS RESULTS