Name:
ID #:

Quiz 3
CMPS 405: Operating Systems
Fall 2015
Dr. Ryan Riley

Imagine that you are programming in a version of Java that only has access to one synchronization primitive: TestAndSet. Assume TestAndSet is a object you can use as follows:

```
// Initialize a new TestAndSet object, set its initial value to false
TestAndSet ts = new TestAndSet(false);
// Set the value to true and return the old value:
boolean oldVal = ts.set(true);
```

Use TestAndSet to build a semaphore class. Fill in your solution to the skeleton code below:

Here is a sample solution. Note that I used TestAndSet to build helper methods, lock and unlock.
Take special note of the way I need to lock and unlock in acquire. Why do I need to unlock and relock while inside the loop?

```
public class Semaphore {
	private int value;
	TestAndSet ts;

	public Semaphore(int value) {
		this.value = value;
		ts = new TestAndSet(false);
	}

	private void lock() {
		while (ts.set(true) == true) {
			Thread.yield();
		}
	}

	private void unlock() {
		ts.set(false);
	}

	public void acquire() {
		lock();
		while (value <= 0) {
			unlock();
			Thread.yield();
			lock();
		}
```

```
            value--;
            unlock();
        }

        public void release() {
            lock();
            ++value;
            unlock();
        }
}
```