

Compile and test using Java 6 on wind.ecs.baylor.edu

SemaSk8r.java (25 points)

LockSk8r.java (35 points)

```
private static final int MAXZAMBONIFICATIONTIME = 100;
private static final int MAXBETWEENZAMBONIFICATIONTIME = 500;
private static final int MAXBETWEENNEWSKATERTIME = 10;
private static final int MAXSKATETIME = 5; // Then try 25
private static final int MAXSKATERS = 2;
```

- Much of the code is mine. Don't consider my code
- Kill any student yields and sleeps
- Make sure that the test code has not been altered
- Verify that the **only** synchronization mechanism they use is the one specified (e.g., if Java locks are specified, then they cannot use semaphores, synchronized blocks, etc.)
- Verify **no** busy wait
- Verify proper handling of interrupts, spurious returns from wait(), etc. -5 bad interrupts/-10 bad waits (only once)
- Run with student settings for 3000
- Run with ConTest 4 times to 2500 with my values. Verify:
 - No test violations
 - Multiple skaters are allowed at one time but no more than 4. Verify with logs.
- Extra credit (+3 for SemaSk8r/+6 for LockSk8r/+10 both) – Give **only** if they ask for consideration. Verify Zamboni gets in after the skaters either skating or waiting ahead of it.

H2OMatch (40 points)

```
private static final int MAXSROUTINETIME = 15;
```

- Most of the code may be mine. Don't consider my code
- Verify **no** busy wait
- No busy wait; only one ballerina should wake up for each iteration (-10 points)
- Verify proper handling of interrupts, spurious returns from wait(), etc. -5 bad interrupts/-10 bad waits (only once)
- Make sure that **my** test() is called
- Run with student settings for 3000
- Run with ConTest 4 times to 2500 with my values. Verify no test violations

For both parts

- Take off points for poor coding practices

-2 unused include/import
-10 for poor commenting
-2 for each unjustified numeric constant
-5 bad submission
-1 for each section of commented code
-5 if violate coding convention