Cosc 4740
Lab 4:
Using Semaphores to avoid race conditions.

Assume that a finite number of resources of a single resource type must be managed. Processes may ask for a number of these resources and will return them once finished. For example, many commercial software packages provide a given number of application licenses, indicating the number of applications that may be run at any given time. When the application is run, the license count is decremented, and when it is no longer in use, the count is incremented. When all licenses are in use, a new instance of the application cannot be started until a previous session has returned a license.

The *thread_semaphore.cpp* program emulates this behavior. The program launches 10 application threads, each of which tries to create a random number (between 1 and 3) of instances by requesting the resource (think license) required to run. The application threads then use them for a short time and return them.

Before you begin, read through the code and then compile and run it before making any modifications.

```
g++ thread_semaphore.cpp –lpthread

./a.out
```

Now run it several times. Notice that we have a maximum number of 5 available resources; the program indicates that more than that are available once all the threads have finished. In this lab you will use semaphores to make sure the program operates correctly.

Follow the directions in the comments of the code to do two things:

Part 1

Identify the race condition and the data involved with it in the increase_count() and decrease_count() functions. Use a mutext lock (binary semaphore) to fix this race condition. (See the code comments for more information)

Part 2

Implement two new functions, increase_S(int count) and decrease_S(int count) using a counting semaphore so that the application threads don't have to "busy wait" for resources. Again, see the comments in the provided code for more information.

NOTE:
You can download files from a web-server via the command line. The following commands will download this lab's files into your current working directory:

```
wget www.cs.uwyo.edu/~seker/courses/4740/labs/thread_semaphore.cpp
wget www.cs.uwyo.edu/~seker/coures/4740/labs/semp.h
```