CSCI 340 FALL 2013

THIS PROJECT **MUST BE DONE INDIVIDUALLY**.

PROJECT 2: Due: Dec. 14th

If you have question(s) regarding submission requirements and programming please email to JLAIQC@GMAIL.COM

DO NOT use synchronized methods (beside the operations on semaphores)
Do NOT use wait( ), notify( ), or notifyAll( ) as monitor methods.
Use only the semaphore class and its methods.

You should keep the concurrency of the threads as high as possible, however the access to shared structures has to be done in a Mutual Exclusive fashion, using a mutex semaphore.

Many of the activities can be simulated using sleep(of a random time) method.

Use appropriate System.out.println( ) statements to reflect the time of each particular action done by a specific thread. This is necessary for us to observe how the synchronization is working.

Document your project and explain the purpose of each semaphore.

Submission similar to project1.

ELLIS ISLAND

Tourists go to Ellis Island to learn about the beginnings of America, and about their ancestors who immigrated to this country.

Every one and a half hours a documentary movie is presented. A movie session contains a presentation period, and the movie. Before the movie session starts, interested visitors wait in the lobby. When the session starts, visitors check if there are available seats, and if yes, they take one of the available seats. If there are no free seats, visitors will wait hoping to catch the next movie session.

Once the next movie session fills, the speaker will signal one visitor on line that will be no future presentation. That visitor will signal another visitor on line and so on, until all waiting visitors will move on.

Visitors that attend the movie session will next listen to the speaker’s presentation. They will take a break (sleep of random time) and next they will wait to watch the movie. After watching the movie they will move on in visiting the museum.

Before the movie starts, a speaker will give a short introduction to the present audience. When the presentation ends, everybody watches the movie.

The speaker waits for the time to start the presentation. If the second movie session is about to start he will first let one of the visitors on line know that will be no next movie session. Next he will also watch the movie. At the end of the day the speaker will leave.

At the end of the day, when the museum is closing, all visitors will group in GroupSize. Next they will leave the museum.

In order to keep track of the time, we need an additional thread, named *clock*. The *clock* will signal when a session starts, when a session ends, and when it is the end of the day. (This will be implemented by having the clock sleep for fixed interval of time) Between movie sessions we should have a break of 15 minutes.

Initial values: theater capacity: 6
Num_visitors: 15
Number_sessions/day: 2
GroupSize: 4

Using Java programming, synchronize the three types of threads, visitor(s), speaker, clock, in the context of the problem. Any wait or signal should be done using semaphores, synchronization between threads should also be implemented through semaphores.