CSE 603 **Programming Assignment 2**
Due Date: Nov 5, 2012
Weight: 8% of your grade

Problem: Dense Matrix Multiplication using CUDA on GPUs: Programming experience.
Description: Implement a dense matrix multiplication algorithm (note that you can do the implementation in many ways) that will evaluate the product of two matrices A and B, to give the product as matrix C. The program needs to have the following features:

1. input matrix generation routines for A and B
2. matrix multiplication to give you C = A x B
3. output matrix verification routine
4. output the time taken to do step 2 (multiplication)
5. input parameters should include the sizes of the two matrices A and B and also the version of input generation (1 or 2) – description in verification section.

The goal of this assignment is to compare the performance of dense matrix multiplication using OpenMP and MPI (using your previous homework results) and GPU with the increase in size of the problem.
You need to have multiple implementations of the GPU program

1) using only global memory with one thread per element of the C matrix
2) using only global memory with one block of thread working on few C elements but having one thread work on only one element of C matrix
3) One thread working on a subset of elements of C matrix
4) Efficient implementation with the use of double buffering to use local memory in the GPU.

In each of the above case state the largest size of matrix multiplication you can do and test your performance upto the max size of the matrix.

Verification: You should synthesize the two input matrices and compare the output matrix. An example is to use the same input matrices where each element of the input matrices is unity. The output matrix can be checked automatically and your program should check for the correctness of the product. You need to have two different routines to create input matrices A and B and have corresponding check of the final output matrix C. This is similar to your previous programming assignment and can be done using the host processor.

Report: Your report should include the following:

1. Complete source code. Include scripts you may have used to run programs with different parameters and also scripts that you may have used to collate results and create graphs (highly recommended to do this to make it easier to finish your work in time).
2. discussion of results to include the following (graphs)
   a. impact on the performance of the algorithm based on
      i. increase in the size of the problem
   b. comparison of performance with your MPI and OpenMP implementations.

Extra Credit (2%): Incorporate a combination of MPI and GPU and compare the performance in terms of increase/decrease of speedup and scalability. Discuss the results as in (2) of report and submit all scripts and source.

Extra Extra Credit (1%): Incorporate a combination of OpenMP, MPI and GPU and compare the performance in terms of increase/decrease of speedup and scalability. Discuss the results as in (2) of report and submit all scripts and source.