CSE494 Topic: Memory & Multithreading Techniques in C++11
Assignment #1
110 possible points, 100 is perfect score

As we have learned in class, std::thread is the way in the new C++11 to create a new thread of execution, where one can pass "work" to be done in the thread's constructor.

Assignment description: you will be creating a simple program which iterates through a sequence of numbers {1…n} and computes the square root of each. You probably should use an outside function to do this. The "sqrt" function is in the header <cmath>. You do not need to print out the results of each (you will be iterating over hundreds of millions of numbers very quickly!).

Important: the number "n" will be read in as argv[1], where argv is defined from:
int main(int argc, char* argv[]) { … }

Assignment directions: you will create 2 "sections" of code.

(15 pts) 1st section: write a serial version of the problem which will iterate over the numbers and compute the square root of each (no need to print or store result anywhere).

(30 pts) 2nd section: write a parallel version of the problem. You will use any number of threads between 2 and 5 (your choice).

(15 pts) Write about how the performance of the parallel version compares with that of the serial version. If the parallel is slower, explain why this may be. Type up your findings in a comment block in the code.

(20 pts) Use the C++11 <chrono> header to measure the amount of time to do the serial and to do the parallel versions of the problem. Share your results in the same comment block (above).

(20 pts) Write a parallel version of the problem that will use the maximum number of threads "allowed" by the system. You will find the max number of threads, and launch each of the threads with a start and end range, which will be the same for each thread.

You will need to modify the original function you had to have 2 parameters instead of 1. (Hint: use std::thread::hardware_concurrency() for finding out the max number of threads).

(5 pts) Your code must use a Makefile and produce an executable called "run".

(5 pts) Your code must use the <thread> header and work on the ASU general server.