Syllabus:
CS 5334/4390 Spring 2016: Parallel and Concurrent Programming

Instructor:
Shirley Moore
Office: CCSB 3.0422
Phone: 915-747-5883
Email: svmoore@utep.edu
Webpage: http://www.cs.utep.edu/svmoore/
Office hours: TR 1:45-2:45pm, others by appointment

Class time and location:
TR 3:00-4:20pm, UGLC 334

Course websites:
http://svmoore.pbworks.com/

Course description:
The goal of this course is to introduce students to the foundations of parallel computing including the principles of concurrency, parallel algorithm design, programming models for shared and distributed memory systems, numerical and non-numerical parallel algorithms, analytical modeling of parallel programs, and debugging and performance optimization of parallel programs. The course will also include material on parallel architecture, new shared-memory programming models (Intel Thread Building Blocks, Cilk), programming models for GPUs, and problem-solving on large-scale clusters using Google's MapReduce. A key aim of the course is for students to gain hands-on experience by writing correct and efficient concurrent and parallel programs in some of the programming models covered in class.

Prerequisites: Programming experience in C/C++ and/or Fortran and/or Java, CS 2302 Data Structures, and CS 3432 Computer Architecture I; or CPS 5401; or permission of the instructor

Textbooks (optional): There will not be a required textbook for the course. The books listed below will serve as references. Additional readings may be assigned on a weekly basis.

- *Parallel Computer Organization and Design* by Michel Dubois, Murali Annavaram, and Per Stenström
- *The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications*, by Clay Breshears. Good overall book on multithreading.
- *An Introduction to Parallel Programming*, by Peter Pacheco. Covers different programming models: pthreads, OpenMP, MPI.
- *Parallel Programming for Multicore and Cluster Systems,* by Thomas Rauber and Gudula Rünger

Learning outcomes:
After successfully completing this course, students should be able to

- DESCRIBE and APPLY concepts of concurrency and parallelism
- EXPRESS concurrent computations and their coordination correctly
- DESCRIBE the major approaches to parallelism used in a large parallel program
- DESIGN and IMPLEMENT a decomposition strategy and parallel algorithm to solve a given numerical or non-numerical problem
- SELECT and APPLY appropriate parallelization constructs to a large program to create a correct parallel version that exploits a state-of-the-art parallel computing environment
- ASSESS the correctness of concurrent and parallel programs,
- APPLY available debugging methods to detect and correct errors an erroneous concurrent and parallel programs
- ASSESS the performance of a parallel program run with different input sizes and numbers of processors
- APPLY performance optimization methods to improve the parallel efficiency and scalability of a parallel program
- DISCUSS current and future trends in parallel architectures and programming models

Course schedule and format:
The course will be taught as a "blended" course with online lectures and assignments provided by PurdueNext and by the instructor. Please see the local instructor's course website at http://svmoore.pbworks.com/ for a detailed and up-to-date schedule for class meetings and assignments. We will use a "flipped classroom" format that will require watching videos, reading, and preparation by class participants prior to class and that will involve group work during class.

Assignment, exams, and grading:
There will be an extra problem on the exams for graduate students that will be optional/extra credit for undergraduates. Some assignments will also have an extra problem for graduate students. Graduate students will also be expected to do a more challenging project. The grading breakdown will be approximately as follows:

Homework and lab assignments	30%
Class preparation and participation	15%
Quizzes	10%
Course exams	30%
Project	15%

Term project:
The term project will consist of:

1. an implementation of a significant parallel algorithm or application
2. a report describing the background and design decisions for the implementation

3. a presentation during the final exam period describing and demonstrating your program

The specific problem can be of your choosing but you must have your topic pre-approved by the instructor. You may work individually on the final project or in teams of up to three people. In the case of group work, you must clearly document the contributions of each team member and carry out the amount and difficulty of work proportional to the size of your team. For example, a team project might implement and compare a set of related algorithms, or compare various parallelization strategies for a given algorithm.

Make-up and late policy:
If you are unable to attend a class or exam or turn in an assignment on time due to a legitimate reason, such as a health problem or accident or pressing family matter, you will be allowed to make up the relevant work. Otherwise, all assignments MUST be turned in by the stated due date and time. NO LATE ASSIGNMENTS WILL BE ACCEPTED.

Accommodations for Students with Disabilities:
If you have a disability and need classroom accommodations, please contact The Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass .

Academic Honesty Policy:
Make sure you understand the UTEP academic honesty policy. Students are encouraged to share ideas, but you must do your own homework and you must write your own code for the projects (you may copy code that is on the course website). If homework or program code is suspected of being duplicated or copied, you will receive an incomplete for the assignment, and your case will be referred to the Dean of Students for adjudication. If the instructor has reason to believe that you have cheated on a quiz or exam, your case will be referred to the Dean of Students for adjudication.