CPS 5401

Introduction to Computational Science

Fall 2015

The University of Texas at El Paso

Shirley Moore, Lecture Instructor

http://www.cs.utep.edu/svmoore/

http://svmoore.pbworks.com/

Natasha Sharma, Lab Instructor

CPS 5401 is an introduction to basic computational science skills including Linux, scientific programming using high level languages, parallel computer architectures, parallel programming paradigms, and numerical libraries.

Course #:	CPS 5401
Course title:	Introduction to Computational Science
Credit hours:	4
Term:	Fall 2015
Time & location:	Lecture: 10:30-11:50am TR, CCSB 1.0510 Lab: 4:00-4:50pm T, Educ 108
Prerequisites:	Instructor approval Recommended: Co-enrollment in MATH 5329
Course fee:	None
Instructor:	Shirley Moore, Lecture Instructor Natasha Sharma, Lab Instructor
Office location:	Moore: Chemistry and Computer Science Building 3.0422 Sharma: Bell Hall 318
Contact info:	Moore: Office phone: 915-747-5883 Email: svmoore@utep.edu Fax: 915-747-5030 WWW: http://www.cs.utep.edu/svmoore/ Sharma: Office phone: 915-747-6858 Email: nssharma@utep.edu Fax: WWW: http://www.math.utep.edu/Faculty/nsharma/public_html/fall2015_5401.html
Office hours:	Moore: 1:00-2:00pm Tuesday, 12:00-1:00pm Thursday, or by appointment Sharma: 4:00pm-5:00pm Thursday or by appointment
Online textbook:	Victor Eijkhout, *Introduction to High-performance Scientific Computing*, http://pages.tacc.utexas.edu/~eijkhout/istc/istc.html

Course Objectives

The course will cover three major aspects of computational science in three parts:

- Part I will consist of a practical introduction to Linux, scientific programming using high level languages, and tools for managing source code and data files.
- Part II will cover parallel computer architecture, parallel programming models, and trends in high performance computing.
- Part III will cover installation and use of parallel libraries for dense and sparse linear algebra.

Grading

Your grade for the course will be based on the following:

- 20% homework
- 30% labs
- 40% midterm and final exams
- 10% class preparation and participation

Attendance Policy

This is a challenging course and attendance is essential for success. Please try not to be absent unless absolutely necessary.

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.

Course Format and Participation

The lecture portion of the class will consist of short lectures interspersed with hands-on interactive activities. Lab assignments will reinforce the lecture material. The lecture and lab exercises will make use of CS Department Linux lab machines, the UTEP Research Cloud, and the Stampede Supercomputer at Texas Advanced Computing Center. Students should be able to login to these resources remotely from a home or office computer. All students should bring a laptop computer to class with which to login to the remote resources. (Please let the instructors know if you do not have a laptop you can bring to class).

Course Topics

1. Linux
    - Shell commands
    - Environment variables and shell programming
    - Job control
    - File system
    - Build systems and source code control
2. Scientific programming languages
    - Compiling and linking
    - Fortran 90
    - C and C++
    - Python and SciPy
3. Computer architecture
    - Cache-based microprocessors
    - Memory hierarchy
    - Shared memory parallel computers
    - Distributed memory parallel computers
    - Hierarchical and hybrid parallel systems
4. Parallel programming paradigms
    - Data and task parallelism
    - Shared memory parallel programming using OpenMP
    - Distributed memory parallel programming using MPI
    - Hybrid parallelization with MPI+OpenMP
5. Dense and sparse linear algebra libraries for
    - Dense linear systems, including least squares and eigensystems
    - Sparse direct methods
    - Sparse iterative methods

Learning Outcomes

1. Manage program and data files on a Linux system
2. Implement basic matrix operations and linear algebra algorithms in the Fortran or C/C++ scientific programming language
3. Implement scientific programming workflows using Python
4. Select the appropriate computer architecture and programming model for a given problem
5. Implement basic matrix operations and numerical linear algebra algorithms in parallel on shared and distributed memory computers using Fortran or C/C++ together with OpenMP (shared memory) and MPI (distributed memory)
6. Call dense and sparse linear algebra library routines correctly from a program written in Fortran or C/C++.