# CPS 5401

## Introduction to Computational Science

### Fall 2017
### The University of Texas at El Paso

**Xianyi Zeng, Instructor**

**Osei Tweneboah, Teaching Assistant**

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 number: | CPS 5401 |
| Course title: | Introduction to Computational Science |
| Credit hours: | 4 |
| Term: | Fall 2017 |
| Time & location: | Lecture: 17:00–18:20pm TR, UGLC 208<br>Lab: 16:00–16:50pm T, UGLC 208 (May change) |
| Prerequisites: | Instructor approval<br>Recommended: Co-enrollment in MATH 5329 |
| Course fee: | None |
| Instructor: | Xianyi Zeng |
| Teaching assistant: | Osei Tweneboah |
| Office location: | Zeng: Bell Hall 202<br>Tweneboah: TBA |
| Contact info: | Zeng:<br>Office phone: 915-747-6759<br>Email: xzeng@utep.edu<br>Website: `http://math.utep.edu/faculty/xzeng`<br>Tweneboah:<br>Email: oktweneboah@miners.utep.edu<br>Website: `http://utminers.utep.edu/oktweneboah` |
| Course website: | `http://math.utep.edu/Faculty/xzeng/2017fall_cps5401` |
| Office hours: | Xianyi: TBA<br>Tweneboah: TBA |
| Online textbook: | Victor Eijkhout, *Introduction to High-performance Scientific Computing*,<br>Website: `http://pages.tacc.utexas.edu/~eijkhout/istc/istc.html` |

### Important

The course website will be updated throughout the semester according to the progress in class. The instructor and the TA will send emails regarding class annoucements. It is your responsibility to check the emails and the course website frequently to keep up to date.

## 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:

- 30% homework.
- 20% lab assignments.
- 20% midterm exam.
- 30% final exam.

## Homework and lab assignments

Late homework or lab assignments will be not accepted under any circumstances. However, among all your homework scores, the lowest one will not contribute to your final grade. The same rule applies to the lab assignments: The lowest score in all your lab assignments will not contribute to the final grade.

## 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 `http://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 computing facilities at the university (to be announced). The 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 instructor know if you do not have a laptop you can bring to class).

## Military statement

If you are a military student with the potential of being called to military service and/or training during the course of the semester, you are encouraged to contact as soon as possible.

## Course drop deadline

**November 3** is the university fall drop/withdrawal deadline. After this date no dropping is allowed.

## Course topics

1. Linux
   - Shell commands
   - Environment variables and shell programming
   - File system and job control
   - Build system and source code control
2. Scientific programming languages
   - Compiling and linking
   - C and C++
   - Python and SciPy
3. Computer architecture
   - Cache-based microprocessors
   - Memory hierarchy
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Hierarchy 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 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 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 C/C++.