CPS 5401 (CRN 12071)

Introduction to Computational Science

Fall 2019
The University of Texas at El Paso

Xianyi Zeng, Instructor

TBD, Teaching Assistant

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

Course number:	CPS 5401 (CRN 12071)
Course title:	Introduction to Computational Science
Credit hours:	4
Term:	Fall 2019
Time & location:	Lecture: 17:00–18:20pm TR, UGLC 208 Lab: 16:00–16:50pm T, BELL 130
Exam dates	Midterm exam: in class TBD, 2019 Term project due: 11:59pm Sunday, December 15, 2019
Drop deadline	Friday, November 1, 2019
Prerequisites:	Instructor approval Recommended: Co-enrollment in MATH 5329
Course fee:	None
Instructor:	Xianyi Zeng
	Office hour: 15:00pm–16:00pm T, 15:00pm–16:50pm R, or by appointment Office location: Bell Hall 202 Office phone: 915-747-6759 Email: xzeng@utep.edu
Teaching assistant:	TBD
	Office hour: TBD Office location: TBD Email: TBD
Course website:	http://math.utep.edu/faculty/xzeng/2019fall_cps5401
Textbook:	None. Slides and handouts will be posted on the course website.
Reference:	Victor Eijkhout, *Introduction to High-Performance Scientific Computing*, 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 the student's 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 consists of a practical introduction to Linux, scientific programming using high level languages, and tools for managing source code and data files.
- Part II covers computer architecture and its implication in writing correct and efficient programs both in serial and in parallel.
- Part III involves parallel programming using both OpenMP and MPI.

Grading

The final grade for the course will be based on the following:

- 30% homework.
- 20% lab assignments.
- 20% midterm exam.
- 30% term project.

Late homework or lab assignments will be not accepted under any circumstances. However, the lowest among all homework scores **and** the lowest among all lab assignment scores will be dropped towards calculating the final grade.

Missed exam **cannot** be made up, either; exceptions can be given only in extraordinary and unavoidable circumstances with reasonable proofs, and with advance notice in written.

The term project will be announced at the beginning of the semester and submission is due at the end of the semester (December 15th, 2019). Aspects of the project will be discussed throughout the semester and the students are encouranged to start building the product as early as possible. Students will form groups of two (or three, in case there are odd number of students) to work on the project. Each group must submit **one** copy of the project; whereas each student must submit a report that specifies individual work.

The letter grade will be guaranteed at the following levels: A (90 - 100%), B (80 - 89%), C (70 - 79%), D (60 - 69%). Depending on circumstances, the thresholds for each of the four letter grades could be lowered, but will be equal among all students. The letter grade F will be given if the final score does not reach the D threshold.

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 the student has 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

The students are required to understand the UTEP academic honesty policy. Sharing ideas are encouraged among the students; but collaboration of any form in the written homework, lab assignment, and exam is strictly prohibited. If the instructor has reason to believe that the students have cheated on the exam, and if the homework or program code is suspected of being duplicated

or copied, the 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 a laptop is not available.

Military statement

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 1 is the university fall drop/withdrawal deadline. Student can still initiate drop after this date, but a grade of W is not guaranteed. Generally, a grade of W will be granted only if the instructor agrees that the drop is due to a valid non-academic reason, such as missing classes or an exam due to medical reasons with proof. If the student wants to withdraw due to poor performance, please do so before the **November 1** deadline.

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`;
   - Scientific libraries for linear algebra tasks.
3. Computer architecture:
   - Cache-based microprocessors;
   - Memory hierarchy;
   - Shared memory parallel computers;
   - Distributed memory parallel computers.
4. Parallel programming paradigms:
   - Data and task parallelism;
   - Shared memory parallel programming using `OpenMP`;
   - Distributed memory parallel programming using `MPI`.

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 linear algebra library routines correctly from a program written in `C/C++`.

Course schedule

Below is a tentative schedule for this course.

Week 01 (08/27, 08/29)	Introduction to Linux. The term project.
Week 02 (09/03, 09/05)	The Unix shell. The `C` programming language.
Week 03 (09/10, 09/12)	The `C` programming language.
Week 04 (09/17, 09/19)	The `C` programming language. Review of numerical linear algebra.
Week 05 (09/24, 09/26)	Scientific libraries. The `C++` programming language.
Week 06 (10/01, 10/03)	The `C++` programming language. GNU Make. CMake.
Week 07 (10/08, 10/10)	The version control system. Midterm exam.
Week 08 (10/15, 10/17)	Single-processor architecture.
Week 09 (10/22, 10/24)	Multicore architecture.
Week 10 (10/29, 10/31)	Parallel computing with OpenMP.
Week 11 (11/05, 11/07)	Parallel computing with `OpenMP`. Parallel computing with `MPI`.
Week 12 (11/12, 11/14)	Parallel computing with `MPI`.
Week 13 (11/19, 11/21)	`Python`.
Week 14 (11/26)	Application of `Python` in SVM.
Week 15 (12/03, 12/05)	Numerical methods for ODEs. Comprehensive review.
Week 16 (12/10)	No final exam.