CPS 5401 (CRN 11938)

Introduction to Computational Science

Fall 2020
The University of Texas at El Paso

Xianyi Zeng, Instructor

CPS 5401 provides 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 11938)
Course title:	Introduction to Computational Science
Credit hours:	4
Term:	Fall 2020
Time & location:	Lecture: 17:00–18:20pm TR, online or COBA 304 Lab: 16:00–16:50pm T, online or COBA 304
Exam dates:	Midterm exam: in class online 10/08/2020 (tentative) Term project due: 11:59pm Sunday, 12/13/2020
Drop deadline:	Friday, October 30, 2020
Prerequisites:	Instructor approval Recommended: Co-enrollment in MATH 5329
Course fee:	None
Instructor:	Xianyi Zeng
	Office hour: 15:00–16:00pm TR Office location: online or COBA 304 Email: xzeng@utep.edu
Teaching assistant:	Yi Xie
	Office hour: TBA Office location: Online Email: yxie3@miners.utep.edu
Course website:	`http://math.utep.edu/faculty/xzeng/2020fall_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`

General course format and Covid-19 information

The majority of our lectures, lab sessions, and office hours will be delivered online using Zoom. Live lecture/lab sessions will be held and recorded at the scheduled time unless otherwise announced; recorded videos will be made available in Blackboard after the live session. Students are not required to join the live sessions except for exams, whose format will be announced separately. Announcements will be made either via email or Blackboard; the students are required to check the emails and the course shell in Blackboard frequently to keep up-to-date. Course materials except the recorded lectures will be posted on the *course website*.

Technology requirements

Course content is delivered via the Internet through the Blackboard learning management system (announcement, grade distribution, and recorded lectures), the course website (other course materials), and Zoom (live sessions). The students need to ensure the UTEP e-mail account is working and that they have access to the Web and a stable web browser. Google Chrome and Mozilla Firefox are the best browsers for Blackboard; other browsers may cause complications. When having technical difficulties, update your browser, clear your cache, or try switching to another browser.

The students need to have access to a computer/laptop as well as a scanner/scanning app; if you wish to join live session discussions and/or virtual office hours, make sure a microphone is available. A webcam is suggested but not required; the instructor would like to create live interactive sessions to know the students better and to provide an opportunity for the students to know their classmates. A Zoom client is required if the student plans to join a live session.

Important: The students are encouraged to contact the UTEP Help Desk (https://www.utep.edu/irp/technologysupport) when they encounter course-irrelevant technical difficulties beyond their scope of troubleshooting.

Network etiquette

As communication online can be challenging sometimes, it's possible to miscommunicate what we mean or to misunderstand what our classmates mean given the lack of body language and immediate feedback. Therefore, please keep these network etiquette guidelines in mind. Failure to observe them may result in disciplinary action.

- Always consider audience. This is a entry-level class for computational sciences; therefore, all communication should reflect polite consideration of other's ideas.
- Respect and courtesy must be provided to classmates and to the instructor at all times. No harassment or inappropriate postings will be tolerated.
- When reacting to someone else's message, address the ideas, not the person. Post only what anyone would comfortably state in a face-to-face situation.
- Blackboard is not a public Internet venue; all postings to it should be considered private and confidential. Whatever is posted on in these online spaces is intended for classmates and instructor only. Please do not copy documents and paste them to a publicly accessible website, blog, or other space.

Course format and attendance

The lecture portion of the class will consist of short lectures interspersed with program demonstrations. Lab assignments will reinforce the lecture material. The lecture and lab exercises will make use of computing facilities at the university to be announced later.

All lectures will be delivered live using Zoom; and recorded lectures will be uploaded to Blackboard. Attendance of the live lectures is not required except for the midterm exam, which will be announced separately. In general, lab sessions will also be delivered live using Zoom; however, seeing that hands-on instructions are typically difficult to conduct online, students may request a face-to-face meeting with the instructor on campus.

In the second half of the course, the students will need to access on-campus computing resource or the CPS computing facility for parallel computations. Any campus visit must follow the university guidelines and it may be canceled anytime.

Covid-19 Accommodation

All students are encouraged to complete the Covid-19 student training at https://covidtraining.questionpro.com. Students are not permitted on campus when they have a positive Covid-19 test, exposure, or symptoms. If the student requests a face-to-face instruction session from the instructor and/or plans to come on campus, please complete the self-screening (http://www.screening.utep.edu) prior to every campus visit.

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 by the end of **December 13th, 2020**. Aspects of the project will be discussed throughout the semester and the students are encouraged 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.

Accommodations for students with disabilities

If the student has a disability and need classroom/exam/homework 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.

Military statement

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

Course drop deadline

October 30 is the university fall drop/withdrawal deadline.
The college of science will not accept drop/withdrawal requests of any form after **October 30**.

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 (08/25, 08/27)	Introduction to Linux. The term project.
Week 02 (09/01, 09/03)	The Unix shell. The `C` programming language.
Week 03 (09/08, 09/10)	The `C` programming language.
Week 04 (09/15, 09/17)	The `C` programming language. Review of numerical linear algebra.
Week 05 (09/22, 09/24)	Scientific libraries. The `C++` programming language.
Week 06 (10/29, 10/01)	The `C++` programming language. GNU Make. CMake.
Week 07 (10/06, 10/08)	The version control system. Midterm exam.
Week 08 (10/13, 10/15)	Single-processor architecture.
Week 09 (10/20, 10/22)	Multicore architecture.
Week 10 (10/27, 10/29)	Parallel computing with OpenMP.
Week 11 (11/03, 11/05)	Parallel computing with `OpenMP`. Parallel computing with `MPI`.
Week 12 (11/10, 11/12)	Parallel computing with `MPI`.
Week 13 (11/17, 11/19)	`Python`.
Week 14 (11/24)	Application of `Python` in SVM or ANN.
Week 15 (12/01, 12/03)	Numerical methods for ODEs. Comprehensive review.
Week 16	Work on the term project. No final exam.