Syllabus for:
COE 322: Scientific Computation

Instructor: Victor Eijkhout

2025 Fall semester

1 Basic information

Number and title	COE 322, Scientific Computation, unique 15875
Instructor	Victor Eijkhout `eijkhout@tacc.utexas.edu`
Co-instructor	Susan Lindsey `slindsey@tacc.utexas.edu`
Time and place	CPE 2.206 TTH 3:30 p.m.–5:00 p.m. (in person)
Instructor office hours	ASE 5.224, Tue 2pm–3pm
Teaching assistant	Geonyeong Lee

2 Rationale

Decades ago, computers were invented to solve mathematical equations, especially in science and engineering. Today, computers are ubiquitous in nearly all aspects of our daily lives, their uses far surpassing their number-crunching origins. The pervasiveness of computers in business, consumer environments – and the resulting revenues – has caused a shift in the types of computer languages and skills taught in computer science departments today. Introductory programming classes are now commonly taught in languages appropriate for industry careers in web development, at the expense of offering scientific computing classes formerly taught for the benefit of technical computing, science and engineering departments at universities as well as the R&D departments in numerous industries. This shift in curricula has occurred even though computers are now fundamental to the conduct of science and engineering, two fields vital for increasing economic productivity, ensuring national security, and addressing many important societal problems.

There are recent signs of a reversal of this trend: new degree programs in computational science, new emphasis in federal spending on computational technologies and R&D,

and even a return of scientific computing classes in some computer science departments, etc. However, the need for well-trained computational scientists and engineers is still urgent.

C++ is currently the most powerful general purpose programming language for developing scientific software: it offers the best features and flexibility for designing robust, high-performance applications. The C++ language is constantly evolving and growing, and new trends and concepts for software development in both languages frequently arise and are rapidly disseminated. The COE 322 course provides a unique opportunity to learn modern usage of this language.

3 Course Aims and Objectives

3.1 Short description

Explores the C++ programming language and the ecosystem around it in the context of developing scientific computing software. Topics include object-oriented design and other aspects of the modern C++ language; program design and testing; source code management; UNIX account management.

3.2 Course Aim

This course teaches intermediate use of C++ (20/23 standard), but it does so as part of a discipline of programming. Notions of program organization, testing, correctness, will be taught along basic matters of syntax.

The course stresses active command of the material taught: each lecture section is accompanied by multiple short programming exercises.

By the end of the course, students will be asked to demonstrate their command of the material by doing a larger programming project. This project is the equivalent of two weeks of programming, and explores some scientific simulation. See the grading section 7 for the relative weight of this project.

3.3 Prerequisites

We presume each student possesses the following basic skills, not directly related to programming:

- Elementary knowledge of Unix account management, e.g. logging into remote machines, file and directory manipulation. Students lacking this knowledge need to do a tutorial by the first week of class.
- Familiarity with basic mathematics (trigonometry, some vector linear algebra)
- Students need to be able to access and use the Zoom and Slack applications for online parts of the course.

In addition we presume that the students have prior knowledge of programming, for instance from COE 301, including the following C++ basics:

- data types
- expressions
- control structures
- scoped constructs such as functions
- arrays

3.4 Learning Outcomes

The learning outcomes of this course are as follows:

- Students will be able to use Unix-based computer systems for programming.
- Students will have active command of the basics of C++20, as demonstrated by a programming project.
- Students will grasp the basics of software project organization, including development and testing strategies.

3.5 Impact on Subsequent Courses in Curriculum

This course is a preparation for subsequent COE courses such as 'COE 332: Software Engineering' and 'COE 379L: Parallel Computing'.

3.6 Relationship of Course to Program Outcomes

This course contributes to the ABET Criterion 3 student outcomes that took effect with the Fall 2019 semester. For more information, see Criteria for Accrediting Engineering Programs, 2019 – 2020 at `https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/`

Student outcomes:

- 5: an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

ABET Program Criteria Achieved:

- P: Computer Usage

3.7 Course Organization

The course is organized as a sequence of topics, each one or two lecture sessions, that cumulatively instill active command of the C++ language. There will be tutorials on the use of Unix, and programmer tools.

Topics taught in this class are, not necessarily in this order:

1. Basic UNIX account management
2. Basic UNIX scripting
3. Object-oriented programming including inheritance
4. more language features: I/O and string manipulation; Pointers; Arrays; Iterators and algorithms; Lambdas.
5. Software development techniques: Test-driven development, Unit testing, debugging.
6. Build systems: Make and CMake
7. Source code and repository management via Git.

4 Instructors' Biographical Information

Victor Eijkhout is a research scientist in the High Performance Computing group at the Texas Advanced Computing Center (TACC). He has a doctoral degree in numerical analysis, and he has longtime experience in programming scientific codes and libraries, especially on parallel computers. His research interests include numerical linear algebra, parallel computing, machine learning, processor performance. He has written several widely used textbooks, as well as many scientific papers.

Susan Lindsey has a degree in computer science from The University of California San Diego. She has an extensive background in the field of High-Performance Computing, with over twenty years at the San Diego Supercomputer Center and now at the Texas Advanced Computing Center. Susan is currently an integral member of User Services at TACC and has contributed to a wide variety of programming projects and technical documentation. She also has previous teaching experience in both C and assembly language programming.

5 Format and Procedures

Class periods will feature both a lecture and lab part, as well as discussions of homework. We encourage a lively participation during the lectures and expect that you participate by asking and answering questions. Active participation makes for a better and more interesting class for you and for us, and allows us to assess your progress and to adjust the class material and/or teaching progress accordingly.

Student progress in this class will be evaluated through homeworks and one major programming project. See section 7. There will be no final examination. We expect timely notification if you cannot fulfill your obligations in time.

6 Other Course Information

Students will be granted access to one of TACC's supercomputer running Unix.

You will need a personal computer, such as a laptop, for in-class labs. This can be used

1. to connect to the class machine, in order to do the labs there; or
2. to do labs locally, in which case the student is responsible for installing any software necessary for a development environment.

You may code your homework and projects on your personal laptop, but the test of correctness of all assignments is whether they compile and execute correctly on the TACC machine.

Lecture slides, the class textbook, and other materials are distributed electronically via this Git repository:

https://github.com/TACC/coe322fall2025

Students will be given access to a TACC machine to do lab exercises. For this they need an account at TACC. If you have ever had an account at TACC, that same account suffices for this class. Otherwise, see

https://www.tacc.utexas.edu/use-tacc/getting-started

https://docs.tacc.utexas.edu/basics/mfa/

https://docs.tacc.utexas.edu/basics/accounts/

We will use Slack for all class discussion and collaboration. An invite link will be sent out over Canvas.

We will use Canvas for class notes, quizzes, announcing homework and publishing grades.

7 Grading

- Once- or twice-weekly graded homework exercises will account for 45 percent of the total grade.
- In-class weekly quizzes will account for 25 percent of the total grade.
- One major programming project will account for 30 percent of the grade.
- There will be no final examination during the finals period. Instead, the programming project is due on the last day of classes.
- Students who actively participate in class may receive a grade bump.

8 Class Policies

Class attendance and participation policy

We expect students to attend and participate in class in accordance with the UT Honor Code. Students are encouraged to ask questions, especially relating to material used in their projects.

Absences, in particular on exam days, should be communicated with the instructors as early as possible.

Homework

Homework solutions will be automatically collected on the date and time due. Homework submitted after the due date will be penalized depending on the degree of lateness. Permission to submit more than a week late needs to be explicitly granted.

Use of AI tools

The goal of this course is not to deliver finished software products, but to acquire and display the programming skills required to produce them. As such, the use of generative AI tools e.g. ChatGPT, Co-pilot, Devin, is discouraged. In judging and grading programming assignments, the instructor may ask clarification from the student to confirm understanding of the submitted code.

Note that it is a violation of university policy to misrepresent work that you submit or exchange with your instructor by characterizing it as your own, such as submitting responses to assignments that do not acknowledge the use of generative AI tools.

Please feel free to reach out to the instructors with any questions you may have about the use of generative AI tools prior to submitting any content that has been substantially informed by these tools.

Academic Integrity

University of Texas Honor Code

The core values of The University of Texas at Austin are learning, discovery, freedom, leadership, individual opportunity, and responsibility. Each member of the university is expected to uphold these values through integrity, honesty, trust, fairness, and respect toward peers and community.

Each student in this course is expected to abide by the University of Texas Honor Code. Any work submitted by a student in this course for academic credit will be the student's own work. Collaborations will be allowed for the course project.

You are encouraged to study together and to discuss information and concepts covered in lecture and the sections with other students. You can give ”consulting” help to or receive ”consulting” help from such students. However, this permissible cooperation should never involve one student having possession of a copy of all or part of work done by someone else, in any electronic form.

Should copying occur, both the student who copied work from another student and the student who gave material to be copied will both automatically receive a zero for the assignment. Penalty for violation of this Code can also be extended to include failure of the course and University disciplinary action.

9 University of Texas Policies

This class observes university guidelines.

Religious Holy Days

By UT Austin policy, you must notify us of your pending absence at least fourteen days prior to the date of observance of a religious holy day. If you must miss a class, an examination, a work assignment, or a project in order to observe a religious holy day, we will give you the opportunity to complete the missed work in a reasonable time after the absence.

Students with Special Concerns

Students with special concerns - be they athletes who might miss class meetings, students with religious observances that interfere with class meetings, or students with disabilities who need special accommodation - are all supposed to notify us about these special needs by the 12th class day which is 13th September 2013.

Use of E-mail for Official Correspondence

All students should become familiar with the University's official e-mail student notification policy. In particulat they should use the new EID-based email format. **The instructors would greatly appreciate it if you use this UT email for your TACC account.** If you already have a TACC account, and it uses a different email account, please change this, or submit a ticket to have it changed.

Students are expected to check e-mail on a frequent and regular basis in order to stay current with University-related communications, recognizing that certain communications may be time-critical. It is recommended that e-mail be checked daily, but at a minimum, twice per week. The complete text of this policy and instructions for updating your e-mail address are available at http://www.utexas.edu/its/help/utmail/1564 .

Documented Disability Statement

Any student with a documented disability who requires academic accommodations should contact Services for Students with Disabilities (SSD) at (512) 471-6259 (voice) or 1-866-329-3986 (video phone). Faculty is not required to provide accommodations without an official accommodation letter from SSD. Please notify us as quickly as possible if the material being presented in class is not accessible (e.g., instructional videos need captioning, course packets are not readable for proper alternative text conversion, etc.).

Other Absences

If you need an extension of deadlines or other accomomdation because of sickness or other unforeseen circumstances, submit (or have submitted) a letter from the Office of the Dean of Students. Requests after the fact ('I have been sick for the last two weeks') will not be honored except in extreme cases.

Behavior Concerns Advice Line (BCAL)

If you are worried about someone who is acting differently, you may use the Behavior Concerns Advice Line to discuss by phone your concerns about another individual's behavior. This service is provided through a partnership among the Office of the Dean of Students, the Counseling and Mental Health Center (CMHC), the Employee Assistance Program (EAP), and The University of Texas Police Department (UTPD). Call 512-232-5050 or visit http://www.utexas.edu/safety/bcal.

Drop Policy

The State of Texas has enacted a law that limits the number of course drops for academic reasons to six (6). As stated in Senate Bill 1231:

> Beginning with the fall 2007 academic term, an institution of higher education may not permit an undergraduate student a total of more than six dropped courses, including any course a transfer student has dropped at another institution of higher education, unless the student shows good cause for dropping more than that number.