Syllabus for:
COE 322: Scientific Computation

Victor Eijkhout

2022 Fall semester

1 Basic information

Number and title	COE 322, Scientific Computation (15010)
Instructor	Victor Eijkhout
	eijkhout@tacc.utexas.edu
Co-instructor	Susan Lindsey
	slindsey@tacc.utexas.edu
Time and place	in person, TTh 3:30–5:00
Office hours	arranged by request
Teaching assistant	TBA
	contact through Canvas or Slack

2 Rationale

Computers were invented over 60 years ago to solve mathematical equations, especially in science and engineering. Over the last 20 years, computers have become ubiquitous in our lives: increasing productivity, enhancing communications and connectivity, ensuring safety, and providing entertainment. The pervasiveness of computers in business and consumer environments – and the resulting revenues – has caused a shift in the computer languages and skills taught in computer science departments today. Introductory programming classes are now commonly taught in Java, and focus on skills needed 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 has occurred even though computers have become increasingly 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 some scientific computing classes in some computer science departments, etc. However, the need for well-trained computational scientists and engineers is still urgent. Fortran, C and C++ remain the most powerful general purpose programming languages for developing scientific software: they offer the best features and flexibility for designing robust, high-performance applications. These languages are constantly evolving and growing, 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 these languages.

3 Course Aims and Objectives

3.1 Catalog description

COE 322 SCIENTIFIC COMPUTATION

Restricted to Computational Engineering majors.

Explores the basic tools needed for developing scientific computing software. These include advanced programming languages (e.g. C, C++, python), object oriented programming and data structures. Subjects may include abstract data types; creation, initialization, and destruction of objects; class hierarchies; polymorphism, inheritance and dynamic binding; generic programming using templates, linked lists, queues, stacks, trees and algorithms such as searching, sorting, and hashing.

Computational Engineering 322 and Statistics and Data Sciences 322 may not both be counted.

Prerequisite: Aerospace Engineering 301 or Computational Engineering 301 with a grade of at least C-.

Note: earlier versions of the catalog listed ASE 201 as possible prerequisite. This is no longer enough. Contact the instructor if you want this to count as sufficient prerequisite.

Designed to accommodate 35 or fewer students.

3.2 Course aim

This course teaches the basic of C++ (2017 standard) and Fortran (2003 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 assume the following basic skills, not directly related to programming:

1. Elementary knowledge of Unix (logging in, file and directory manipulation) is assumed. Students lacking this knowledge need to do a tutorial by the first week of class.
2. Familiarity with basic mathematics (trigonometry, some vector linear algebra) is assumed.
3. Students need to be able to access and use Zoom and Slack for online parts of the course.

We assume that the students have prior knowledge of programming, for instance from COE 301, including the following C++ basics:

1. data types
2. expressions
3. control structures
4. scoped constructs such as functions
5. 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++17 and Fortran2003, 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 later courses such as

- COE 332: Software Engineering
- SDS 335/374: 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 sessions, that cumulatively instill active command of the languages C++ and Fortran. There will be tutorials on the use of Unix, and programmer tools.

Topics taught in this class are:

1. Basic unix scripting
2. Object-oriented programming including inheritance
3. I/O and string manipulation.
4. Pointers
5. Test-driven development, Unit testing, debugging.
6. Arrays: iterators and algorithms.
7. Lambdas.
8. selected features of the Standard Temmplate Library.

The Fortran language, including its object-oriented features, will then be taught in accelerated tempo, since the students already have command of the basic principles.

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 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 a strong track record 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

6.1 Computers

Students will be given access to a Unix computer located at TACC.

For in-class labs, a personal computer such as a laptop is needed. 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 compilers or a development environment.

It is allowed and possible to do homework on a personal machine, but the test of correctness of labs and homework submissions is whether they compile and execute correctly on the TACC machine.

6.2 Electronic resources

1. Lectures slides, a textbook, and other materials are distributed electronically through a Git repository.
   `https://github.com/TACC/coe322fall2022`
   Learning the bare basics of Git, that is the `clone` and `pull` commands, is recommended.
2. 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` and `https://portal.tacc.utexas.edu/tutorials/multifactor-authentication`
3. We will use Slack for discussion and collaboration. An invite link will be sent out over Canvas.
4. Canvas will largely be used only for announcing homework and publishing grades.

7 Grading Procedures

- There will be once or twice weekly graded homework exercises, counting for 70 percent of the total grade.

- There will be one major programming project counting for 30 percent of the grade.
- There will be no final examination during the finals period. The programming project is due on the last day of classes.

Participation in class may cause your grade to be rounded up.

8 Formal and informal policies

Covid-19 specific policies

This class observes university guidelines.

In particular, note:

- Instruction will be in-person subject to changes.
- Classes will be recorded as much as possible and viewable through UT's Zoom/Canvas integration.
- Instructors will be masked and ask students to do so too.

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.

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 an opportunity to complete the missed work within a reasonable time after the absence.

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 the form of an e-mail, an e-mail attachment file, a diskette, or a hard copy.

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.

During examinations, you must do your own work. Talking or discussion is not permitted during the examinations, nor may you compare papers, copy from others, or collaborate in any way. Any collaborative behavior during the examinations will result in failure of the exam, and may lead to failure of the course and University disciplinary action.

Other University Notices and Policies

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 to Students

All students should become familiar with the University's official e-mail student notification policy. It is the student's responsibility to keep the University informed as to changes in his or her e-mail address. 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.).

Please notify us as early in the semester as possible if disability-related accommodations for field trips are required. Advanced notice will permit the arrangement of accommodations on the given day (e.g., transportation, site accessibility, etc.).

a notice that students with disabilities may request appropriate academic accommodations from the Division of Diversity and Community Engagement (DDCE), Services for Students with Disabilities (SSD) at http://ddce.utexas.edu/disability.

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.

Emergency Evacuation Policy

Occupants of buildings on the UT Austin campus are required to evacuate and assemble outside when a fire alarm is activated or an announcement is made. Please be aware of the following policies regarding evacuation:

- Familiarize yourself with all exit doors of the classroom and the building. Remember that the nearest exit door may not be the one you used when you entered the building.
- If you require assistance to evacuate, inform us in writing during the first week of class.
- In the event of an evacuation, follow the instructions of class instructors.

Do not re-enter a building unless you're given instructions by the Austin Fire Department, the UT Austin Police Department, or the Fire Prevention Services office.