Cornell Bowers C·IS
College of Computing
and Information Science

CS 4414: Systems Programming

Cornell Bowers CIS Department of Computer Science

Fall 2025 Course Syllabus

Course Website: https://www.cs.cornell.edu/courses/CS4414/2025fa/

General Information

Faculty name: Ken Birman

Email: ken@cs.cornell.edu
Office Hours: After class and Thursday, 8:30am – 10:30am Gates 435

Course staff: The web site will list course staff once hiring decision have been made. CS 4414 and CS 5416 share the same staffing, but the TA responsible for the recitation lectures differs (as does content), as explained below.

Credits and Grading Options

- **Credits:** 4.0 credits
- **Grading:** Letter grade or S/U only; no audit.
- **Recitation:** Attendance is required. The recitation often covers material that was not presented in the T/R lectures, and the prelims test that material. Cornell sometimes uses the term "discussion section" for the recitation, but our recitations cover actual material – they are not really just discussions.
- **Co-Meets:** Shares prelims and lectures with CS 5416, but projects differ. The required recitation for CS 5416 meets separately from the one for CS 4414. Although the CS 5416 recitations cover any material needed for prelims, those recitations also cover additional content used in the CS 5416 projects.

Please note:

- Students who take CS 4414 for a grade are not permitted to enroll in CS 5416.
- Under NYS policy, graduate students can only count CS 5416 towards degree requirements; CS 4414 is not eligible for that purpose.
- While the undergraduate and graduate versions of the course share lectures and prelims, the final projects are different and the CS5416 recitations are light on C++ content, and instead cover some topics not explored in CS 4414.

Prerequisites

Fall 2024 version of CS 3410 (or later), or written permission of the instructor. Specific CS 3410 topics you must be comfortable with include (1) prior experience with C or C++ and memory pointers, malloc and free; (2) experience with threading; (3) computer architecture and design features, notably for multicore NUMA servers.

Time and Location

To be determined. In Fall 2024, CS 4414 met on Tuesdays and Thursdays from 2:55pm-4:10pm in Uris Hall G01 for a total of 28 sessions and had 14 recitation discussion sessions on Fridays from 2:55pm-4:10pm.

Prelims and Projects

50% of your grade will be based on two evening prelims. We do not have makeups, but do accommodate people who have a medical excuse for the evening of an exam. Students who will be out of town interviewing are still required to take the prelim at the same time as the students in Ithaca.

The remaining 50% of your grade is based on assignments and multi-stage projects; the very first projects don’t count for very much, whereas the last submission for the last project counts double.

Final project submission date.

We start grading final projects on the last day of classes, but allow you to submit until midnight on the published date for the CS 4414 final (recall that we do not actually have a final exam). **If the registrar doesn't list CS 4414 on the finals schedule, the due date will be midnight on the first day of finals week.**

Enrollment Information and Questions

If you have questions about enrolling in this course, please start by reviewing the Bowers CIS enrollment policies and waitlist information available on the Bowers CIS Courses Help website. If you can't find the answer to your question, submit a ticket on that same page for assistance. In the event of a delay enrolling, you should still attend all classes and do all assigned work.

Websites

The web site for the course is always kept up to date, so even though we do use Canvas, Ed Discussions and Gradescope, always use the web site for information about the syllabus, prelim rooms, etc. We use Ed Discussions for important announcements, so set it up to notify you daily and read them.

Course Description

CS 4414 exposes students to programming applications at the systems level and to the operating systems abstractions that these applications depend on. It then builds on this foundation to look at systems issues that shape the performance and reliability of modern ML and AI applications, such as "chat bots" and question-answering AIs. **We do not expect students to understand how these ML and AI tools work, in a mathematical sense.** Instead our focus is on how the execute, where components run and how they talk one another, how they interact with big-data storage, and how they leverage accelerators such as GPU.

Across this spectrum of scenarios, both for classical system services and for modern AI and ML use cases, there is a great deal of commonality. Students learn to make design choices guided by performance, hardware, security and other systems properties. Operating systems abstractions covered include process and memory management, file systems and storage, networking, threads and multiprocess concurrency along with synchronization abstractions including locks and condition variables, and security abstractions for isolation and authorization. Students gain experience with C/C++ programming, major command-line tools and techniques for debugging, instrumenting and tuning applications.

Our focus on CS++ may surprise students who have only used PyTorch in their AI and ML courses, but in fact is quite standard for the AI area: although PyTorch, Tensor Flow and other quick AI-builder frameworks are popular ways of *prototyping* new applications, and they do leverage GPU accelerators, any host-compute aspects of the solutions are often inefficient when compared to C++ versions that use the same algorithms. The industry as a whole favors C++ for compute-intensive AI and ML tasks in which host-compute plays a substantial role.

Learning Objectives

- Proficiency programming in C++ 20, using Visual Studio Code IDE on Linux (Ubuntu) demonstrated through successful completion of hands-on assignments.
- Proficiency with Linux commands and bash programming demonstrated through successful completion of hands-on assignments.
- Ability to write multithreaded code that leverages the full performance of modern NUMA servers demonstrated through successful completion of hands-on assignments that focus on speeding up code by using multicore parallelism.
- Interpretation of parallelism in many forms, and ability to create parallel solutions to practical computing problems, to implement them correctly in C++, and to debug and optimize solutions. Demonstrated through a mix of exam performance and ability to use these ideas when creating hands-on assignments.
- Ability to work with the monitor synchronization pattern in C++, and a deeper recognition of associated issues such as race conditions, critical sections, coordination through the consumer producer and readers-writers models, deadlocks, livelocks. Demonstrated through a mix of exam performance and ability to use these ideas when creating hands-on assignments.
- Appreciation of how these skills carry over to creating AI and ML solutions that perform well (meaning, they use computing platforms efficiently, don't waste electric power, and are able to respond to tasks quickly).

Summary of Skills Assessed

- Proficiency in C++ 20 and Ubuntu Linux.
- Understanding of the concepts underlying modern systems, concurrency and parallelism, and how to extract the maximum performance from the hardware.

- Skill using performance debugging tools to identify bottlenecks and performance-limiting architectural choices.
- Skill in developing alternative implementation and architectural designs that can overcome these bottlenecks.
- Practice applying these ideas in substantial multi-week, multi-stage projects that often require out of the box thinking and learning-by-experience.
- Understanding the performance issues seen in modern AI and ML applications.

Projects

Although the first weeks of CS 4414 include self-contained homework assignments intended to give students a chance to come up to speed on our development environment, the course quickly shifts to a pattern of multi-week, multi-stage projects. In fact our final project is due after prelim 2, and for this reason CS 4414 has no final exam.

Our philosophy is that CS 4414 lectures teaches concepts that we can test on prelims, the course also has a substantial experiential learning component in which we will guide you to build larger C++ programs and then apply the performance and design ideas covered in class. We also expect you to to understand performance bottlenecks and possible ways to eliminate them and to exercise that understanding by improving performance for AI / ML applications that we will provide to you. Thus project grading is a substantial part of the overall grade for CS 4414. Please note that graduate students who take CS5416 share the same lectures as CS 4414, but undertake an even more ambitious project (one still focused on the performance of an AI / ML system, but with that system running on a cloud and including even more components).

Course Materials

There are no required textbooks for this course. The following is a list of optional, but useful, references for different parts of the course:

[1] Randal E. Bryant and David R. O'Hallaron, Computer Systems: A Programmer's Perspective, Third Edition, Pearson, 2016. Please note that this is a large and expensive textbook, and we only draw on a few chapters.

[2] Bjarne Stroustrup, A Tour of C++ (2nd Edition), Addison-Wesley Professional, (July 9, 2018). There are many other C++ books if you find this one too terse.

[3] Linux. Linux has comprehensive online documentation and LLMs such as ChatGPT and Claude do an excellent job of answering questions and giving examples.

Method of Assessing Student Achievement

Your performance in CS 4414 is determined by assessing your classroom learning and understanding via prelim exams and your skills via assigned homeworks and projects (the latter being larger multi-step assignments that span several weeks). CS 4414 is curved, hence if the entire class does unusually poorly on an exam, or unusually well, the curve still brings the overall distribution of scores in line with that for prior offerings. One consequence is that the mapping from score to letter grade will depend on the curve and cannot be predicted just by knowing your numerical grades on each prelim and assignment.

The course is conformant with the official grading system at Cornell. Because we curve the class, the overall median final letter grade is typically in the A-/B+ range and relatively few students receive letter grades lower than B-. Having said this, we do need to acknowledge that some do receive C's or even D's. Historically, these have generally been students who didn't attend class or who failed to complete project assignments.

Schedule

The course schedule is available on the course website: https://www.cs.cornell.edu/courses/cs4414/2025fa/, on the "schedule" tab. Please note that until the 2025 web sites are opened up by the department, in August, the only version available to you will be the 2024fa site. While there is substantial similarity, because we extensively revamped the course after CS 3410 was redesigned in Fall 2024, many lectures will have changed and some are completely new. The old slide decks and videos will certainly be useful for studying or catching up if you miss a lecture, but it would not be possible to do well in CS 4414 without attending lectures.

Course Management Policies and Expectations

Late policy

Coding assignments and related deliverables must be submitted electronically by uploading a zip file to Gradescope. **No other formats will be accepted!**

Each student may use slip days when submitting assignments. You do not need to request them. Each slip day provides an automatic 24-hour extension, but also has a price: each slip day “costs” 5pts relative to the maximum possible score. After a maximum of three slip days (72 hours), uploads are no longer permitted because we often hand out a solution set or other materials needed for the next step of a multi-stage assignment.

Regrade Policy

Addition errors in the total score are always applicable for regrades. Regrades concerning the actual solution should be rare and are only permitted when there is a significant error. Please only make regrade requests when the case is strong and a significant number of points are at stake. Regrade requests should be submitted online via Gradescope within 72 hours after we release the graded material.

SDS Accomodations

The Student Disability Service, SDS, is the one-stop office for discussing health issues and other accommodation requests at Cornell. Large courses often have a number of people with SDS accommodation letters. We are committed to respecting such letters. Nonetheless, some letters cover multiple courses and for this reason may include language that is not directly applicable to CS 4414.

All CS 4414 students must take both prelim exams and complete all homework assignments. If an SDS letter gives the impression that you will be excused from prelims or homeworks this is a misunderstanding of the wording. Instead, Professor Birman will work with you to find a way to complete all the required work so that you can be assessed on exactly the same basis as the students who do not have an SDS accommodation letter.

Some students have difficulty concentrating in large exam rooms. CS 4414 exams always set aside one room for students who need a quieter space, with its own proctor to supervise and answer any questions. If you lack an SDS letter but feel that taking your exam in this quieter space would be beneficial, speak to Professor Birman and he will grant access on a per-case basis.

SDS accommodation letters often include a non-binding recommendation that the professor grant 50%, 75% or 100% “extra time” on exams. However, CS 4414 exams *are not time-pressured*. A typical CS 4414 prelim is designed as a 75m exam, but to avoid a sense of time pressure our proctors allow all students to stay for 2 ½ and this can sometimes even extend to 3 hours if the room is available and the TA is able to work that late. All students can benefit from this form of extra time.

A few final remarks:

- If you experience any access barriers in this course, such as with printed content, graphics, online materials, or any communication barriers, reach out to the course staff or SDS.
- If you need immediate accommodation, please speak with Professor Birman after class or send an email message to me and SDS at sds_cu@cornell.edu.
- If you have, or think you may have a disability, please contact Student Disability Services for a confidential discussion: sds_cu@cornell.edu or visit the SDS website to learn more.

Collaboration Policy, Academic Integrity, Use of AI Tools

The work you submit in this course is expected to be the result of your individual effort only. Your work should accurately demonstrate your understanding of the material. The use of a computer in no way modifies the standards of academic integrity expected under the University Code of Academic Integrity, which you can review at http://cuinfo.cornell.edu/aic.cfm.

You are encouraged to study together and to discuss information and concepts covered in lecture with other students. You can give “consulting” help to or receive “consulting” help from other students. Students can also freely discuss basic computing skills or the course infrastructure. However, this permissible cooperation should never involve one student having possession of or observing in detail a copy of all or part of work done by someone else, in the form of an email, an email attachment file, a flash drive, a hard copy, or on a computer screen.

Students are not allowed to seek consulting help from online forums outside of Cornell University. Students are not allowed to use online solutions (e.g., from Course Hero, Chegg, etc.) from previous offerings of this course. We are comfortable with study groups and peer consulting, but no student should show any other student their actual code.   Questions should be of the “how should I think about this?” variety, not “how would you fix this bug?”

CS 4414 does allow the use of modern AI-based tools (“CoPilots”), such as Microsoft CoPilot (built into Visual Studio Code), ChatGPT or Claude. Obviously, such tools can generate C++ code for small tasks such as implementing a double-linked queue. However, after our first small assignments, all the CS 4414 projects are much more ambitious than what an AI can solve.

If you do include a snippet of code from an AI, for example to show you how to call a Linux file read system call, you do not need to document that. But if an AI writes a procedure for you and you include that into your solution, you should fully document that the method is code obtained from such-and-such a source. It is never acceptable to use an AI to solve large portions of any assignment.

Some students worry that unless they get AI help, their work will not be the very best in the class.   Indeed, not every person will be able to achieve the very best solution on every assignment. But remember that for us, YOUR learning is the goal. The AI was already trained – we are trying to teach you something, not to test your skill in getting the AI to spit that information out.   Doing your own work is far more valuable than just obtaining a solution elsewhere.

We always award partial credit and often you will get a good grade even if your work is not flawless. Conversely, if you try to pass off work that is not your own we have a high likelihood of noticing and you will face an academic integrity proceeding, which is far more of a problem than a B- letter grade.

Wellness, Mental Health

Your health and wellbeing are important. Nobody can do their best work when they are struggling with other issues that preoccupy them. Moreover, coding is best done when well-rested, without too much caffeine (or other substances), when eating a normal diet, and interleaved with other activities It is often more effective to take a break than to spend hour after hour staring at a bug.

There are services and resources at Cornell designed specifically to bolster undergraduate, graduate, and professional student mental health and well-being. Remember, your mental health and emotional well-being are just as important as your physical health. If you or a friend are struggling emotionally or feeling stressed, fatigued, or burned out, there is a continuum of campus resources available to you: https://mentalhealth.cornell.edu/ get-support/support-students. Help is also available any time day or night through Cornell's 24/7 phone consultation (607-255-5155). You can also reach out to Prof. Birman, your college student services office, your resident advisor (if applicable), or Cornel Health for support. Also, refer to the resource guide compiled by the members of Body Positive Cornell, EARS, Reflect, and Cornell Minds Matter.

Additional Resources

Other related resources can be found here:

- Study Resources
- Writing Resources
- Library Liaisons