ECE 459: Programming for Performance (Winter 2018)

Department of Electrical & Computer Engineering
University of Waterloo

About the Course

Undergraduate Calendar Description "Profiling computer systems; bottlenecks, Amdahl's law. Concurrency: threads and locks. Techniques for programming multicore processors; cache consistency. Transactional memory. Streaming architectures, vectorization, and SIMD. High-performance programming languages."

Prerequisites ECE 254 or SE 350 (or equivalent); Level at least 4A Electrical Engineering or Computer Engineering or Software Engineering.

Or, to put that in less formal terms: remember when we talked about semaphores and mutexes and all of that? If not, well, `https://github.com/jzarnett/ece254/tree/master/lectures`. Also, I really hope you feel comfortable with programming. This isn't the place for you if you're not feeling confident in programming. Really. Does `void*` scare you? If it does, flee while you still can!

Brief Overview

Many modern software systems must process large amounts of data, either in the form of huge data sets or vast numbers of (concurrent) transactions. This course introduces students to techniques for profiling, rearchitecting, and implementing software systems that can handle industrial-sized inputs. These techniques will enable you to design and build critical software infrastructure, especially in an age of Big Data.

While you may have seen some of these ideas in the context of operating systems (ECE254/SE350/CS350) and concurrency (CS343), this course gives you tools to make code run faster. The focus in OS is understanding and implementing the primitives; our focus is on using them effectively.

We will sometimes see implementation details that you need to get right to write certain applications, but as with any university-level course, this course focusses more on the concepts than magic invocations, so that you can continue to apply the basic ideas after the technologies inevitably change.

General Information

Course Website As is standard, information will be posted on Learn in various folders. But you will see all the magic at `https://github.com/jzarnett/ece459`. The primary source for course materials is github.

Two Sections This year, due to student demand, there are two sections of ECE 459. Section 002 was created to meet demand from Software Engineering students. Lecture content will be largely similar, and assignments and tests will be identical.

Lectures 001	MF	8:30 – 9:50	STC 0060
Lectures 002	TTh	11:30 –12:50	DWE 3522
Tutorials		Exist in the schedule, but we won't use them	

Midterm exam to be announced. It will be 60 minutes in length.

Final exam to be announced on or after 2 February; it will be scheduled by the RO as per usual.

Schedule Oddities: Reading week is 19-23 February. No classes on 30 March (Good Friday); 4 April will follow the Friday schedule. 3rd and 4th year classes don't happen on the day of Capstone Project Symposium (14 March for SE students, 21 March for ECE students) although that doesn't affect any lectures in this course directly. We're also aware that IRS is a thing.

Final Exam: The final exam dates are 9 April - 24 April. Partway through the term, the registrar's office will announce the exact date, time, and location of the exam. It could fall at any time in this period. Note that student travel plans are *not* considered an acceptable reason for missing an exam. When it is announced, please alert your instructor immediately if you have a conflict.

About Prof. Zarnett. I graduated from the Computer Engineering program at Waterloo (under the previous curriculum), and have since earned my Master's Degree (also at Waterloo) and my P.Eng. license. For the last 4+ years I have also been teaching here at UW and the other courses I have taught and worked on include ECE 150, ECE 155, ECE 254, ECE 290, ECE 356, and MTE 241.

In addition to being your instructor, I work full time in software engineering in industry. It's my job to help you learn, so contact me if you feel the need. The vast majority of the week, I will not be on campus, so stopping by my office is a very inefficient way to find me. I'm more than happy to answer questions by e-mail, and if they can't be answered by e-mail, we can set up an appointment for office hours.

About Prof. Lam. I'm excited to be teaching half of the SE class of 2018. Coincidentally, I'll have taught roughly half of the SE students currently in the program at this point. And, I am also halfway through my term as Director of Software Engineering. But I hope that you get full value out of this course.

As many of you know, I can often be found in DC2597C. Feel free to try and drop by anytime. I'll be happy to talk to you if I'm in. You can ensure availability by sending me email. Professoring is my full time day job. My research interests focus on using compiler and static analysis techniques to ensure program correctness. We'll touch on that at some points in this term.

Lab Instructor. We have a lab instructor this term. The LI is responsible for technical matters related to the assignments: ecgit, git in general, administering the system where you need to run your labs..

Stephen Li
Email: `stephen.li@uwaterloo.ca`

About the Teaching Assistants. Teaching assistants can help you with the course material, including assignments and exams. They will do most of the lab marking.

Teaching Assistants:

Tejinder Singh
Email: `t42singh@uwaterloo.ca`

Kamal Lamichhane
Email: `klamichh@uwaterloo.ca`

Husam Suleiman
Email: `hsuleima@uwaterloo.ca`

Rollen Sandeep D'Souza
Email: `rs2dsouz@uwaterloo.ca`

Reference Material

You might find this book useful but it is not required.

> Software Performance and Scalability: A Quantitative Approach, Henry H. Liu, John Wiley & Sons, 2009.

Evaluation

This course includes assignments, a midterm, and a final examination. Too many fourth-year courses have projects. This tends to make the end of term insane. We won't contribute to that problem too much.

Assignments	60% (4 at 15% each)
Midterm	10%
Final exam	30%

A grade of INC (Incomplete) is to be assigned if you do not attempt all assignments. If the assignments are not completed within 4 months of the end of the course to the satisfaction of the course instructor, the INC (Incomplete) grade will be converted to FTC (Failure to Complete). If the assignments are completed successfully within the required timeframe, the overall grade will be calculated using the rules for late assignment submission. Furthermore, any submitted assignment must be an honest attempt – submitting an empty project or similar is not adequate.

The University rules say if you miss the final exam, without an acceptable reason, your grade in the class will be DNW - Did Not Write. This is very undesirable. Show up for the final exam.

Assignments. Since this course has "programming" in the title, you will be expected to write code for these assignments. Here is a projected list of assignments for this course. We plan to have 4 assignments as below:

1. Manual parallelization for servers with Pthreads and asynchronous I/O;
2. Using compiler-provided automatic parallelization and OpenMP;
3. GPU programming with OpenCL;
4. Load balancing.

Assignment hand-in will be done via `git` using the university provided ecgit service.

Your assignment code will be checked for plagiarism using MOSS (Measure Of Software Similarity) as well as some manual checks. You may request to opt out of the automatic screening by sending a formal written letter to your instructor explaining why; a meeting will then follow to discuss the subject with the instructor.

You'll have at least two weeks to do each assignment. Trying to place the assignment due dates in the term is hard; We try to avoid FYDP deadlines, not place them on top of midterms, keep them vaguely in sync with the course material covered so far, and finally we're not allowed to make the due date for them during reading week or after the end of term.

Tutorials. This class has tutorials scheduled. 4th year students basically never attend tutorials so there's no point in using them, right?

Exams. Exams will be open-book, open-notes, calculators without communication capability, no communication devices.

Group work. You may discuss assignments with others, but we expect each of you to do each assignment independently. Acceptable collaboration includes discussing ideas and structures with others, as well as helping others debug their code. If your code is too close in structure to someone else's code, you are going to have a problem. The best way to avoid such problems is by (1) not sending your code around; and (2) not writing down anything beyond general notes (pseudocode) about other peoples' code. We will follow UW's Policy 71 if I discover any cases of plagiarism (and we have).

We want to emphasize that we take the issue of plagiarism very seriously, and so does the University of Waterloo. If you are uncertain about this subject, please seek some guidance. There are many resources available to you. You can check the university policies, talk to the course instructor, ECE/SE undergrad office, et cetera.

Or, let's sum this up in two short instructions:

1. Acknowledge the work of others.
2. If you are uncertain, ask!

Lateness. Also known as "Grace Days". You have 4 days of lateness to use on submissions throughout the term. Each day you hand in something late consumes one of the days of lateness. The fifth day of lateness causes your lowest assignment mark to be halved, while the sixth day causes both assignment marks to be halved. If you hand in something and you have more than 6 days of lateness, I'll start converting marks to 0 and dropping the associated late days. You don't get any credit for unused late days.

For example, you may hand in A1 one day late, A2 two days late, A3 1 one day late, and everything else on time. Or you can hand A2 four days late, if you hand in everything else on time. Finally, if you hand in A1 3 days late, A2 1 day late, A3 3 days late, and everything else on time, We will either give you a 0 for A1, leaving you with 4 late days, or give you a 0 for A2, leaving you with 5 late days and causing your mark for A1 to be halved. We'll choose the option which gives you more marks.

Re-marking If you believe that your grade on an a written, submitted deliverable (e.g., a midterm exam question) is incorrect or unfair, you may ask that it be re-marked. To request that a question be re-marked, you will need to submit your request on a sheet of paper, in writing, to one of the instructors [PL or JZ]. You may submit it in person, or ask an administrative assistant at the Electrical & Computer Engineering undergraduate office to put it in PL or JZ's mailbox. Please do not hang around outside JZ's office hoping to find him (this is really inefficient). Hanging around PL's office may work better. Please do not submit it to a TA or lab staff.

When you submit your request, it should include the following: (1) Your name and student ID number; (2) a clear indication of which question or part of the deliverable is to be re-marked; and (3) an explanation of why you believe the grade assigned was incorrect.

If you received the marked version of the deliverable back (e.g., the midterm exam), please submit that alongside your written request. Staple them (not paperclip, not some sort of origami) together so they do not get separated.

We will accept items for re-marking any time before the final exam, including as we are arriving to the room where the final exam is to be written. Be forewarned, when a deliverable is being re-marked, your grade could go up, it could stay the same, or it could go down. We will notify you of the outcome and attempt to return the deliverable you submitted (if any).

Extra Credit In this class, there will be no opportunities to earn extra credit. Make-up assignments or examinations will not be offered under any circumstances.

Attendance & Illness Personal opinion on attending classes: it is usually a good idea to attend lectures. That said, attendance is not taken and not graded.

Some advice Professor Gebotys gave long ago: If you are tired, go sleep at home. Sleeping in the lecture doesn't work; you will get poor quality of sleep and you won't learn the material while you're asleep, either.

During the term, you may need arrive late to a class or leave partway through, because of job interviews. This is not a problem, as long as you are not disruptive when arriving/departing.

If you feel ill, you should seek appropriate medical attention. If you miss an exam for health reasons, you need a verification of illness form. Forms can be completed by the physicians at Health Services. If you anticipate missing a deliverable deadline or an examination for a non-medical reason, you should contact your instructor as soon as you are aware of the problem. Given sufficient notice, alternate arrangements may be possible. Alternate arrangements are rare and at instructor discretion.

Laptop and Device Policy The human visual system has evolved to perceive saber-toothed tigers in the savannah. Fortunately, tigers are rare in Waterloo, Ontario (Geese, on the other hand...). Unfortunately, your classmates are still human and hence their attention will be drawn to flashing lights (or Facebook, or movies, or video games) in their peripheral vision. We'd like to encourage everyone to be respectful of their classmates and to not distract them.

Wise use of computers and the Internet can be helpful for fully engaging in class. You might want to try out some syntax, or you might want to look up C++ constructors, or you might want to verify your instructor's somewhat outrageous-sounding claim.

To support the benefits of the Internet while reducing distractions, we will adopt the following policy in this class. We are asking that the first 4 rows of class be text-oriented: if using a device, use a command prompt or text editor, maximized to the whole screen. Paper is always good, of course. Mac OS X and UNIX command prompts are probably your best bet; for those of you on Windows, you can use the Windows Subsystem for Linux. From the command prompt, you can use compilers and text-mode web browsers (w3m, lynx, links/elinks, etc....) tmux may also be helpful in managing multiple terminal sessions. Being proficient with the terminal is a highly-useful skill for a Software Engineer.

We acknowledge that lectures are not always engaging. Instead of distracting screen content, we recommend non-distracting ways of tuning out, like doodling on paper (while taking notes), or doing homework. (we also recommend passing notes to each other instead of talking).

If you need to sit towards the front of the class and use a GUI program, then please discuss with your instructor to register yourself as an exception. If you sign up for the exception list, we'll ask to you agree to not display games, videos, or social media on your screen (unless it is part of the class).

Enforcement is a sensitive issue, especially given the existence of exceptions. We are primarily asking each of you to respect the policy on your own. But, if you see someone with games, videos, or social media in the terminal zone, you can politely bring it up with them.

tl;dr: paper or text-oriented programs in first 4 rows of class.

University Policies

Academic Integrity In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check `www.uwaterloo.ca/academicintegrity/` for more information.

Grievance A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, `adm.uwaterloo.ca/infosec/Policies/policy70.htm`
If in doubt, contact the department's administrative assistant, who will provide further assistance.

Discipline A student is expected to know what constitutes academic integrity (see above section) to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor,

or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, www.adm.uwaterloo.ca/infosec/Policies/policy71.htm . For typical penalties check Guidelines for the Assessment of Penalties, see
www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm .

Appeals A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals)
www.adm.uwaterloo.ca/infosec/Policies/policy72.htm.

Privacy Questions about the collection, use, and disclosure of personal information by the University, should be directed to the Freedom of Information and Privacy Coordinator, Secretariat, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1. The email address of the Freedom of Information and Privacy Coordinator is fippa@uwaterloo.ca. See also University of Waterloo Policy 19: Access to and Release of Student Information; Information and Privacy.
https://uwaterloo.ca/secretariat/policies-procedures-guidelines/policy-19

Note for Students with Special Needs The AccessAbility Services (formerly known as OPD) located in Needles Hall, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the AccessAbility Services office at the beginning of each academic term.