ECE 718
Special Topics in Computation
Compiler Design for High Performance Computing

COURSE OUTLINE

Please refer to course website for updated information.

CALENDAR DESCRIPTION

This course introduces advanced topics in compiler design for high performance computing. It covers the basic principles of compiler design and shows how compilers can enable fast applications on single-thread and multi-core processors. The course also explores existing domain-specific and generic compilers that exploit parallelism and memory locality, based on compile-time and runtime analysis.

SCHEDULE And MODE OF DELIVERY

McMaster expects to be fully in-person in the 2025/26 academic year. Please check with instructor and/or avenue to learn for schedule and mode of delivery.

Lecture: TBD
Location: TBD

INSTRUCTOR

Dr. Kazem Cheshmi
Email: cheshmi@mcmaster.ca
Office: ITB-A217
Office Hours**:** After the class or by appointment

COURSE OBJECTIVES

By the end of this course, students should be able to:

- Learn how to write fast code.
- Analyze code using local and global techniques.
- Build a basic domain-specific compiler.
- Understand compiler transformation and capabilities.
- Perform code optimization at both thread and vector instruction levels.
- Work on a collaborative coding project.

ASSUMED KNOWLEDGE

To perform the course project, you will need medium to advanced-level coding skills in C++ and Python. Additionally, it is necessary to possess knowledge of data structures and object-oriented design, as well as a good grasp of computer architecture. High-performance computing background, including parallel computing skills, will be important, although you will receive a brief overview of relevant topics during the course. Collaborative tools, like Git, will be heavily relied upon for course assignments and projects, so it is essential to have Git knowledge.

COURSE MATERIALS

Textbooks:

- *Kumar, Vipin, Ananth Grama, Anshul Gupta, and George Karypis. Introduction to parallel computing. Vol. 110. Redwood City, CA: Benjamin/Cummings, 1994.*
- Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2020). *Compilers: principles, techniques and tools*.

Additional Reading:

The course will go over novel compilers and recent research papers. Some of the papers are:

- Blumofe, R. D., Joerg, C. F., Kuszmaul, B. C., Leiserson, C. E., Randall, K. H., & Zhou, Y. (1996). Cilk: An efficient multithreaded runtime system. *Journal of parallel and distributed computing*, *37*(1), 55-69.
- Cheshmi, K., Kamil, S., Strout, M. M., & Dehnavi, M. M. (2017, November). Sympiler: transforming sparse matrix codes by decoupling symbolic analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-13).

COURSE OVERVIEW

Week	Topic
1	Course Introduction, logistics, overview
2	High-performance computing, matrix-matrix multiplication case study, performance measurement
3	Code optimization techniques, Bentley rules, bit-level optimization
4	Code representation, local analysis, dead code elimination
5	Local value numbering, global analysis, dynamic compilers
6	Single-Instruction Multiple-Data (SIMD), vectorization, Advanced Vector Extensions (AVX)
7	Thread-level parallelism, parallel programming for multi-core processors
8	LLVM Introduction
9	Domain-specific compilers, building schedules, code generation, Halide compiler
10	Domain-specific compilers, Runtime systems, cache-oblivious algorithms, Cilk compiler
11	Domain-specific compilers, Compilers for Sparse Matrix operations, organizing runtime information in sparse codes, discussing Sympiler approach, code specialization
12	Project presentations and reflections
13	Project presentations and reflections

At certain points in the course, it may make good sense to modify the schedule. The instructor may modify elements of the course and will notify students accordingly (in class, on the course website).

ASSESSMENT

Component	Weight	Approximate Due Date
Project	50	The end of semester
3-4 Seminars	50	Multiple Due dates throughout the semester
Total	100 %	

The course project gets students involved with building a new compiler/programming language. The project needs coding in either C++ or/and Python. Presentations will be related to different stages of the project.

CONDUCT EXPECTATIONS

As a McMaster graduate student, you have the right to experience, and the responsibility to demonstrate, respectful and dignified interactions within all of our living, learning and working communities. These expectations are described in the *Code of Student Rights & Responsibilities* (the "Code"). All students share the responsibility of maintaining a positive environment for the academic and personal growth of all McMaster community members, **whether in person or online**.

It is essential that students be mindful of their interactions online, as the Code remains in effect in virtual learning environments. The Code applies to any interactions that adversely affect, disrupt, or interfere with reasonable participation in University activities. Student disruptions or

behaviours that interfere with university functions on online platforms (e.g. use of Avenue 2 Learn, WebEx or Zoom for delivery), will be taken very seriously and will be investigated. Outcomes may include restriction or removal of the involved students' access to these platforms.

COPYRIGHT AND RECORDING

Students are advised that lectures, demonstrations, performances, and any other course material provided by an instructor include copyright protected works. The Copyright Act and copyright law protect every original literary, dramatic, musical and artistic work, **including lectures** by University instructors.

The recording of lectures, tutorials, or other methods of instruction may occur during a course. Recording may be done by either the instructor for the purpose of authorized distribution, or by a student for the purpose of personal study. Students should be aware that their voice and/or image may be recorded by others during the class. Please speak with the instructor if this is a concern for you.

ACADEMIC ACCOMMODATIONS OF STUDENTS WITH DISABILITIES

Students with disabilities who require academic accommodation must contact Student Accessibility Services (SAS) at 905-525-9140 ext. 28652 or sas@mcmaster.ca to make arrangements with a Program Coordinator. For further information, consult McMaster University's *Academic Accommodation of Students with Disabilities* policy.

ACADEMIC ACCOMMODATION FOR RELIGIOUS, INDIGENOUS OR SPIRITUAL OBSERVANCES (RISO)

Students requiring academic accommodation based on religious, indigenous or spiritual observances should follow the procedures set out in the RISO policy. Students should submit their request to their Faculty Office ***normally within 10 working days*** of the beginning of term in which they anticipate a need for accommodation or to the Registrar's Office prior to their examinations. Students should also contact their instructors as soon as possible to make alternative arrangements for classes, assignments, and tests.

EXTREME CIRCUMSTANCES

The University reserves the right to change the dates and deadlines for any or all courses in extreme circumstances (e.g., severe weather, labour disruptions, etc.). Changes will be communicated through regular McMaster communication channels, such as McMaster Daily News, A2L and/or McMaster email.

RESEARCH ETHICS

The two principles underlying integrity in research in a university setting are these: a researcher must be honest in proposing, seeking support for, conducting, and reporting research; a

researcher must respect the rights of others in these activities. Any departure from these principles will diminish the integrity of the research enterprise. This policy applies to all those conducting research at or under the aegis of McMaster University. It is incumbent upon all members of the university community to practice and to promote ethical behaviour. To see the Policy on Research Ethics at McMaster University, please go to http://www.mcmaster.ca/policy/faculty/Conduct/ResearchEthicsPolicy.pdf.

www.eng.mcmaster.ca/ece