DALHOUSIE UNIVERSITY
Faculty of Computer Science

THERE WILL BE NO CONVERTING OF FINAL GRADES TO PASS/ILL THIS SEMESTER.
ALL LETTER GRADES WILL BE FINAL!

CSCI-4118/6105 — Algorithm Engineering
Course Syllabus

Instructor Information

Instructors:	Chris Whidden	**E-mail:** `cwhidden@dal.ca`
Office Hours:	TBA	
Class Meeting Time:	WF 14:35-15:55	Studley MCCAIN ARTS&SS 2016
Lab Meeting Times:	M 11:35-12:55	Studley MCCAIN ARTS&SS 2170
Course Homepage:	`https://dal.brightspace.com`	
Microsoft Team:	FCS CSCI4118/6105 2024 Winter	Join Code: ugaa5k3
Teaching Assistant:	TBA	
Course Representative:	TBA	

Important Dates

1. Final Withdrawal Date without academic penalty: February 6, 2024
2. Final Withdrawal Date with financial penalty: March 6, 2024
3. Munro Day (no classes): February 2, 2024
4. Winter Study Break (no classes): February 19–23, 2024
5. Good Friday (no classes): March 29, 2024
6. Deadlines: Draft Project Proposal on January 26, Revised Project Proposal on February 9, Project Presentations April 5–April 9, Project Report April 9.
7. Labs: Four labs on January 29, February 12, March 4 and March 18.

Course Description

This course presents techniques and methodologies for Algorithm Engineering. Students will learn best practices for developing efficient algorithms and easy-to-use, well-tested, and high-performance implementations of algorithms for real world use. Practical algorithm concerns include maintaining numerical precision, optimizing for realistic rather than worst case inputs, cache efficient computing for processing big data, analysis of parallel and GPU algorithms, predicting the results of design choices and running experiments to verify those choices. The course includes lectures, hands-on labs exploring each aspect, reading assignments and discussions, and a course project providing the opportunity to gain hands on experience with algorithm engineering techniques.

The types of algorithms studied will be fairly broad and may include numerical optimization (e.g. linear programming and geometric algorithms), graph-based algorithms (e.g. shortest paths), sorting and searching, combinatorial algorithms (dynamic programming and fixed-parameter algorithms), huge memory algorithms (I/O efficient, succinct data structures) and a little bit of machine learning (e.g. data concerns like unbalanced classes, noisy data, rare events, etc).

Class Format and Course Communication

- Content will be delivered using labs, lectures and a course project.
- The course project may be completed individually or in groups of 2-3 students.

- Some computers may be available to be borrowed from the FCS helpdesk for labs if needed but in general students are expected to install and run software on their own computers during labs.
- Students must ask the instructor permission before recording class lectures.
- Course announcements will be posted to Brightspace. It is the student's responsibility to check their Dal e-mail and Brightspace on a daily basis. To access your Dal e-mail account please see: `https://www.dal.ca/dept/its/o365/services/email.html`
- You can ask questions and discuss course content with your peers using Microsoft Teams. If you are unfamiliar with Teams please see: `https://dalu.sharepoint.com/sites/its/SitePages/teams.aspx`. The instructors will answer questions posted on Teams when they are able (e.g. during office hours).

Course Rationale

- Students will learn about the practical implementation, tuning and experimentation of algorithms on real data and systems.
- This course complements other Faculty of Computer Science courses that focus on developing efficient algorithms in theoretical terms or touch on empirical efficiency in specific subdomains such as bioinformatics or machine learning.
- Algorithm Engineering is an emerging area of research that is important for numerous scientific domains, as such this course will help attract and train honours undergraduates and graduate students, particularly for the Algorithms and Bioinformatics cluster.
- Algorithm Engineering is also important for the practical design and implementation of algorithms in industry. Students who complete this course will be better able to predict the impact of design choices as well as determine and justify when such optimizations are warranted.

Learning Outcomes

- Differentiate between the goals of worst-case algorithm analysis and algorithm engineering.
- Explain the steps of running a program on a modern computer system and provide an example, illustrating its phases including the time and resources required.
- Develop abstract but realistic models for problems using multiple different types of models.
- Explain why designing algorithms with consideration of practical constraints is important in the production of high-quality software.
- Identify commmon assumptions that lead to incorrect programs (E.g. numerical precision, data integrity, user error) and apply strategies for avoiding such errors.
- Identify and exploit data patterns and features to design faster algorithms for particular sets of input.
- Understand how algorithm techniques like dynamic programming and fixed-parameter tractability can be applied to compute or approximate theoretically difficult (E.g. NP-hard) problems given some assumptions on the input.
- Explain common caching strategies and the impact of memory management on running an implementation of an algorithm.
- Implement multiple variations of an algorithm with different principles of memory management and contrast the results.
- Differentiate between common computational models (E.g. single CPU, multiple CPUs, GPU) and explain the algorithm design constraints that impact each model.
- Predict the running time impact of changes to an algorithm. Implement the changes and run experiments to evaluate the practical impact.
- Apply algorithm engineering concepts to design and implement an algorithm with practical implementation in mind. Design and run experiments to evaluate the practical implementation. Report the results of a practical implementation in a professional manner.

Assessment Criteria

1. Quizzes (10%)
2. Labs (20%)

Lab	Title	Value	Date	Description
1	Understanding Data	5%	Jan 29	Students will run and optimize algorithms for different input data subsets such as sorting nearly-sorted lists and pre-processing the agreement forest problem with leaf and cluster reductions
2	Time-space trade-offs	5%	Feb 12	Students will speed up a dynamic programming algorithm using extra space and probe the behaviour of a succinct data structure with different time-space tradeoffs.
3	Cache Efficiency	5%	Mar 04	Students will run sorting and searching algorithms on increasingly large data sets with and without large memory data structures.
4	Cascading Errors	5%	Mar 18	Students will run and correct code that fails due to floating point imprecision, such as addition and multiplication, and in geometric algorithms like detecting convex hulls.

   - **Students are expected to install and run software on their own computer during labs.**
   - Labs are due at 11:59pm, on the Monday night one week after the lab occurs.
   - **Late labs will only be accepted with submission of a student declaration of absence form (SDA) following the SDA rules. If an SDA is submitted prior to the lab due date via Brightspace (maximum twice) then you will receive a 3 day extension on the lab. See the SDA rules for more information.**
   - Labs need to be submitted electronically using Git.
   - Labs are intended to give hands on experience with algorithm engineering concepts. Students will be provided with software or source code based on the lecture material, modify the code and run experiments to probe the behaviour of the software after modification. Students will report their results and discuss the ramifications.
   - Undergraduate and graduate students will have the same labs but graduate students will be expected to assess each topic in greater detail and may have additional questions to answer.
   - Labs are expected to take approximately 1-1.5 hours of lab time and be completed by undergraduate lab students in a lab period. The analysis component will take an additional roughly 1-1.5 hours for graduate students.
3. Project Proposal and Literature Review (20%)
   - Students will write a 1-2 page proposal and literature review outlining an algorithm engineering project they will complete either alone or in groups of 2-3 students. The scope of the project should match the size of the group and the role of each student in a group project must be clearly explained.
   - Implementation note: Group size may be adjusted for enrollment, e.g. a larger class may scale by requiring groups and/or allowing larger groups.
   - Students in a group will receive the same mark; if difficulties arise this can be changed at the instructor's discretion.
   - Students will plan implementations or modifications of an algorithm, explain the data and system they will use to test the modifications, and describe their hypothesis of the results.
   - Graduate students are expected to propose a substantial project demonstrating 2-3 algorithm engineering techniques. Graduate students will be expected to include a literature review of 3-5 papers on their chosen topic and the expected real world benefit of applying algorithm engineering techniques to this topic. Graduate students are expected to implement an algorithm from scratch with algorithm engineering in mind or substantially modify an existing complex system or algorithm.
   - Undergraduate students are expected to propose a project demonstrating at least 1 algorithm engineering technique. Undergraduate students will summarize a research paper or code documentation and may modify an existing algorithm or plan experiments to probe the behaviour of an existing system.
   - Project Guidance

  - Example project topics will be provided and students will have an opportunity to discuss ideas with the instructor and other students.
  - The initial proposal is a draft. Students will have the opportunity to respond to feedback on the proposals and clarify concerns with the instructor before proposal marks are finalized. The instructor may recommend that some students with related topics group together.
  - As we progress through the class there will be discussions about how the current topic applies to some of the projects
- Proposals will be evaluated based on:
  - topic interest and scientific merit
  - completeness and clarity of literature review
  - feasibility of the approach
  - expected contribution and hypothesis of results

4. Project Implementation and Report (30%)
   - Students will implement their proposed algorithm, run their experiments and report the results in a 3-5 page report.
   - A reproducible workflow for running their experiments must be provided
   - Projects reports will be evaluated based on:
     - execution of the proposed plan
     - clarity and completeness of the analysis
     - discussion of the results in context with prior work
     - reproducibility of the results
     - explanation of any difficulties and limitations of the results
     - clarity and professional nature of the report
   - Note that projects will **not** be evaluated based on "success" or "failure" of the original plan.
5. Project Presentation (20%)
   - Students will report on the results of their project and lessons learned. Presentation length will vary from 5 to 15 minutes depending on the number of projects in the course.
   - Presentations will be evaluated based on:
     - project motivation and hypothesis
     - description of prior work
     - description of the methods
     - description of the results
     - discussion of the results in context with prior work and proposed future work

Notes

- A minimum C grade is required in this course if it is core to an undergraduate FCS degree, or if it will be used as a prerequisite for a subsequent CSCI course.
- A minimum B- grade is required in this course for a graduate FCS degree.
- As of 2019, students who receive a grade lower than C in the same required undergraduate CS course twice, will be dismissed.
- The grade conversion scale in Section 17.1 of the Academic Regulations, Undergraduate Calendar will be used for undergraduate students.
- The grade conversion scale in Section 7.7 of the Faculty of Graduate Studies Regulations, Graduate Calendar will be used for graduate students.
- Graduate and undergraduate students will complete the same assessments and use the same criteria but graduate students are expected to complete much more extensive course projects.
- The instructor reserves the right to adjust a student's evaluation criteria, with the student's consent if the instructor deems that an adjustment is warranted.

Student Declaration of Absence

The Student Declaration of Absence policy shall apply as noted above. Late labs, proposal, or reports may receive a 3 day extension. Quizzes and presentations are not subject to SDAs. https://www.dal.ca/campus_life/safety-respect/student-rights-and-responsibilities/academic-policies/student-absence.html

Academic Standards

Failure to properly attribute sources in your work will be treated as an academic standards issue and points may be deducted for not following citation requirements. For example, forgetting to quote text taken from other sources, failure to include in-text citations, or a failure to include required information in the citations or references. Please see the resources on proper citation provided by the Dalhousie Writing Center (https://dal.ca.libguides.com/c.php?g=257176&p=5001261).

Please note that if it appears that the error was made with intent to claim other people's work as your own such as a lack of both citations and references, an allegation of plagiarism will be submitted to the Faculty Academic Integrity Officer, which could result in consequences such as a course failure.

Recommended Texts and Resources

- Matthias Müller-Hannemann and Stefan Schirra. Algorithm Engineering: Bridging the Gap Between Algorithm Theory and Practice. Springer-Verlag, 2010.
- The lecture slides and course materials will be posted on the learning management system (Brightspace).

Prerequisites

Pre-requisite (CSCI 4118 only): CSCI 3110: Design and Analysis of Algorithms

Tentative Schedule on Next Page:

Tentative Schedule and List of Topics

Date	Item	Topic	Textbook
Jan 10	Lecture 1:	Administrivia and Motivation	1.1
Jan 12	Lecture 2:	Building Blocks of Algorithm Engineering	1.2
Jan 17	Lecture 3:	Modeling: Modeling problems, graph-based models	2.2, 2.31
Jan 19	Lecture 4:	Modeling: Fixed-parameter tractability, approximation	2.3.2–2.3.5
Jan 24	Lecture 5:	Modeling: Understanding data for reduction and conversion	2.4
Jan 26	Lecture 6:	Algorithm design: Simplicity and scalability Draft Project Proposals due	3.2,3.3
Jan 29	Lab 1:	Lab: Understanding Data	
Jan 31	Lecture 7:	Algorithm design: Time-space trade-offs	3.4
Feb 02	No Lecture	Munro Day	
Feb 07	Lecture 8:	Algorithm design: Understanding data for machine learning	2.4
Feb 09	Lecture 9:	Project Discussion and Brainstorming Revised Project Proposals due	
Feb 12	Lab 2:	Lab: Time-space trade-offs	
Feb 14	Lecture 10:	Algorithm analysis: Worst-case, average-case and amortized analysis; realistic input models	4.2, 4.3, 4.5
Feb 16	Lecture 11:	Algorithm analysis: Smoothed analysis	4.4
Feb 19–23		WINTER STUDY BREAK	
Feb 28	Lecture 12:	Algorithm analysis: Computational testing, representative operation counts and experimental study of asymptotic performance	4.6–4.8
Mar 01	Lecture 13:	Computer models: Memory hierarchies, success stories	5.2, 5.5
Mar 04	Lab 3:	Lab: Cache Efficiency	
Mar 06	Lecture 14:	Computer models: Parallel computing	5.3, 5.4, 5.6
Mar 08	Lecture 15:	Implementation: Correctness and efficiency	6.2, 6.3
Mar 13	Lecture 16:	Implementation: Flexibility and ease of use; efficiency of implementation	6.4–6.6
Mar 15	Lecture 17:	Implementation: Numerical precision and geometric algorithms	6.7
Mar 18	Lab 4:	Lab: Cascading Errors	
Mar 20	Lecture 18:	Experiments: Planning, set-up and running experiments	8.2, 8.5
Mar 22	Lecture 19:	Experiments: Test data generation and test data libraries	8.3, 8.4
Mar 27	Lecture 20:	Experiments: Evaluating and reporting experimental results	8.6, 8.7
Mar 29	No Lecture	Good Friday	
Apr 03	Lecture 21:	Case studies: precise running time prediction	
Apr 05	Lecture 22:	Review / Project Presentations	
Apr 08	Lecture 24:	Project Presentations	
Apr 09	Lecture 24:	Project Presentations	

Responsible Computing Policy

Usage of all computing resources in the Faculty of Computer Science must be within the Dalhousie Acceptable Use Policies (`https://www.dal.ca/dept/university_secretariat/policies/information-management-and-t acceptable-use-policy-.html`) and the Faculty of Computer Science Responsible Computing Policy. (`https://www.dal.ca/content/dam/dalhousie/pdf/faculty/computerscience/policies-procedures/fcs_policy_local.pdf`)

Use of Plagiarism Detection Software

All submitted assignment may be passed through a plagiarism detection software, such as the Moss Software Similarity Detection System (https://theory.stanford.edu/~aiken/moss/), or similar systems. If a student does not wish to have their assignments passed through plagiarism detection software, they should contact the instructor for an alternative. Please note, that code not passed through plagiarism detection software will necessarily receive closer scrutiny. https://cdn.dal.ca/content/dam/dalhousie/pdf/dept/university_secretariat/policy-repository/OriginalitySoftwarePolicy.pdf

Use of Artificial Intelligence Tools

You may use AI-driven tools to assist you in learning but remember that the objective is for you to acquire these competencies and outcomes in this course. You are responsible for all work you produce, whether assisted by an AI-driven tool or not. You must acknowledge all tools used to assist you. If applicable, you must provide links to chat logs. If the work you produce is suspected to misrepresent your own competencies, you may be asked to complete a supplemental assessment to evaluate your mastery of course outcomes.

Culture of Respect

Every person has a right to respect and safety. We believe inclusiveness is fundamental to education and learning. Misogyny and other disrespectful behaviour in our classrooms, on our campus, on social media, and in our community is unacceptable. As a community, we must stand for equality and hold ourselves to a higher standard.

What we all need to do [1]:

1. **Be Ready to Act:** This starts with promising yourself to speak up to help prevent it from happening again. Whatever it takes, summon your courage to address the issue. Try to approach the issue with open-ended questions like "Why did you say that?" or "How did you develop that belief?"
2. **Identify the Behaviour:** Use reflective listening and avoid labeling, name-calling, or assigning blame to the person. Focus the conversation on the behaviour, not on the person. For example, "The comment you just made sounded racist, is that what you intended?" is a better approach than "You're a racist if you make comments like that."
3. **Appeal to Principles:** This can work well if the person is known to you, like a friend, sibling, or co-worker. For example, "I have always thought of you as a fair-minded person, so it shocks me when I hear you say something like that."
4. **Set Limits:** You cannot control another person's actions, but you can control what happens in your space. Do not be afraid to ask someone "Please do not tell racist jokes in my presence anymore" or state "This classroom is not a place where I allow homophobia to occur." After you have set that expectation, make sure you consistently maintain it.
5. **Find or be an Ally:** Seek out like-minded people that support your views, and help support others in their challenges. Leading by example can be a powerful way to inspire others to do the same.
6. **Be Vigilant:** Change can happen slowly, but do not let this deter you. Stay prepared, keep speaking up, and do not let yourself be silenced.

[1] Source: Speak Up! ©2005 Southern Poverty Law Center. First Printing. This publication was produced by Teaching Tolerance, a project of the Southern Poverty Law Center. Full "Speak Up" document found at: http://www.dal.ca/dept/dalrespect.html Revised by Susan Holmes from a document provided April 2015 by Lyndsay Anderson, Manager, Student Dispute Resolution, Dalhousie University 902.494.4140 lyndsay.anderson@dal.ca www.dal.ca/think.

University Statements

This course is governed by the academic rules and regulations set forth in the University Calendar and the Senate.
https://academiccalendar.dal.ca/Catalog/ViewCatalog.aspx?pageid=viewcatalog&catalogid=117&loaduseredits=False

Territorial Acknowledgement

Dalhousie University is located in Mi'kma'ki, the ancestral and unceded territory of the Mi'kmaq. We are all Treaty people.

Internationalization

At Dalhousie, 'thinking and acting globally' enhances the quality and impact of education, supporting learning that is "interdisciplinary, cross-cultural, global in reach, and orientated toward solving problems that extend across national borders."
https://www.dal.ca/about-dal/internationalization.html

Academic Integrity

At Dalhousie University, we are guided in all of our work by the values of academic integrity: honesty, trust, fairness, responsibility and respect. As a student, you are required to demonstrate these values in all of the work you do. The University provides policies and procedures that every member of the university community is required to follow to ensure academic integrity.
(read more: http://www.dal.ca/dept/university_secretariat/academic-integrity.html)

Accessibility

The Student Accessibility Centre is Dalhousie's centre of expertise for matters related to student accessibility and accommodation. If there are aspects of the design, instruction, and/or experiences within this course (online or in-person) that result in barriers to your inclusion please contact:
https://www.dal.ca/campus_life/academic-support/accessibility.html
for all courses offered by Dalhousie with the exceptionof Truro.

Conduct in the Classroom — Culture of Respect

Substantial and constructive dialogue on challenging issues is an important part of academic inquiry and exchange. It requires willingness to listen and tolerance of opposing points of view. Consideration of individual differences and alternative viewpoints is required of all class members, towards each other, towards instructors, and towards guest speakers. While expressions of differing perspectives are welcome and encouraged, the words and language used should remain within acceptable bounds of civility and respect.

Diversity and Inclusion — Culture of Respect

Every person at Dalhousie has a right to be respected and safe. We believe inclusiveness is fundamental to education. We stand for equality. Dalhousie is strengthened in our diversity. We are a respectful and inclusive community. We are committed to being a place where everyone feels welcome and supported, which is why our Strategic Direction prioritizes fostering a culture of diversity and inclusiveness (Strategic Priority 5.2).
(read more:http://www.dal.ca/cultureofrespect.html)

Student Code of Conduct

Everyone at Dalhousie is expected to treat others with dignity and respect. The Code of Student Conduct allows Dalhousie to take disciplinary action if students don't follow this community expectation. When appropriate, violations of the code can be resolved in a reasonable and informal manner—perhaps through a restorative justice process. If an informal resolution can't be reached, or would be inappropriate, procedures exist for formal dispute resolution.
(read more: https://cdn.dal.ca/content/dam/dalhousie/pdf/dept/university_secretariat/policy-repositc
Code%20of%20Student%20Conduct%20rev%20Sept%202021.pdf)

Fair Dealing Policy

The Dalhousie University Fair Dealing Policy provides guidance for the limited use of copyright protected material without the risk of infringement and without having to seek the permission of copyright owners. It is intended to provide a balance between the rights of creators and the rights of users at Dalhousie. (read more: https://www.dal.ca/dept/university_secretariat/policies/academic/fair-dealing-policy-.html)

Originality Checking Software

The course instructor may use Dalhousie's approved originality checking software and Google to check the originality of any work submitted for credit, in accordance with the Student Submission of Assignments and Use of Originality Checking Software Policy. Students are free, without penalty of grade, to choose an alternative method of attesting to the authenticity of their work, and must inform the instructor no later than the last day to add/drop classes of their intent to choose an alternate method. (read more: https://cdn.dal.ca/content/dam/dalhousie/pdf/dept/university_secretariat/policy-repository/OriginalitySoftwarePolicy.pdf)

Student Use of Course Materials

These course materials are designed for use as part of the CSCI courses at Dalhousie University and are the property of the instructor unless otherwise stated. Third party copyrighted materials (such as books, journal articles, music, videos, etc.) have either been licensed for use in this course or fall under an exception or limitation in Canadian Copyright law. Copying this course material for distribution (e.g. uploading material to a commercial third party website) may lead to a violation of Copyright law.

Learning and Support Resources

Please see https://www.dal.ca/campus_life/academic-support.html