University of Alberta

CMPUT 303 & CMPUT 403
Fall 2023

Instructor: Zachary Friggstad
Office: 3-06 Athabasca Hall
E-mail: zacharyf@ualberta.ca
I generally do not respond to emails during the weekend.

Teaching assistants:

- Mason Bly (mdbly)
- Akif Chowdhury (mdakif)
- Johnson Wu (jiache17)

Lectures: Tuesday/Thursday 9:30am - 10:50am in ETLC E 2001
Seminars (403 only): Thursday at 1:00pm - 1:50pm in ED 255
Office Hours: See eClass for exact times. Some are in person, some are online, some are hybrid.

COURSE CONTENT

CMPUT 303 - Algorithmics in Practice
This course is focused on algorithmic problems, where a solution involves properly understanding a written description, designing an efficient algorithm to solve the problem, and then correctly implementing the solution. Students will use previous knowledge in algorithms, data structures, and mathematical reasoning to solve problems in addition to learning new algorithms and data structures. Lectures are shared with CMPUT 403. Credit cannot be obtained for both CMPUT 303 and CMPUT 403. Prerequisites: One of CMPUT 201 or CMPUT 275, CMPUT 204.

CMPUT 403 - Algorithmics in Competitive Programming
This course is focused on algorithmic problems that can be solved within at most several hours by well-prepared people, where a solution involves properly understanding a written description, designing an efficient algorithm to solve the problem, and then correctly implementing the solution. Students will use algorithms, data structures, and mathematical reasoning to solve problems. Lectures are shared

with CMPUT 303. CMPUT 403 covers additional material relevant to advanced programming contests. Credit cannot be obtained for both CMPUT 303 and CMPUT 403. Prerequisites: One of CMPUT 201 or CMPUT 275, CMPUT 204, and any 300-level Computing Science course, or consent of the instructor.

Course Overview
Get credit for solving programming challenges!

This course is for anyone interested in designing and implementing efficient algorithms, such as (but certainly not limited to) those taught in CMPUT 204, 275, and 304. The problems you solve are the same type as you would see on an ACM-ICPC programming contest. As a bonus, mastering these challenges will help you with the technical portions of job interviews! It should be noted that many problems you are required to solve in either CMPUT 303 or CMPUT 403 will be noticeably harder than typical programming job interview questions.

Though the course is loosely associated with our problem solving club, participation in an actual contest is not required. Your marks are earned individually.

This is largely a **self-guided** course where students take the foundational understanding of algorithms learned from other courses and apply them to implement efficient solutions to algorithmic problems. The “lectures” are more like meetings: the instructor will introduce a topic that the student should become familiar with but most of the lecture time will be spent working through examples of problems - class participation is key to the success of these discussions!

Students will have to learn about algorithms not covered in other undergraduate courses in order to solve some problems. As such, this course is best suited for those who are willing to invest **considerable time and effort outside of the meetings**. But the goals of this course are different than in CMPUT 204 where you went through full proofs of correctness for algorithms. We just want you to effectively use them as tools to solve other problems: understanding and being able to provide full proofs of correctness is not required in this course.

Students are expected to know a programming language that is supported by Open Kattis. This site supports C++, C, Java, Python2, Python3 among other languages. The prerequisite courses do cover some of these languages, but if you are taking this course and have only received transfer credit for the prerequisites then please understand that you will be required to use one of these languages. Note, some of the more challenging problems with strict time limits may be difficult to solve with a slower language like Python or Java.

If you are wondering if this course is appropriate for you, go ahead and try a few problems on that site. In this course, you will solve many problems using different specific algorithms (eg. network flow, shortest paths, convex hulls) and broader algorithmic paradigms (eg. dynamic programming, divide and conquer, greedy, data structures).

CMPUT 303 vs CMPUT 403 - An Overview

- CMPUT 403 students will be required to use a few more advanced algorithmic concepts or data structures than CMPUT 303 students
- CMPUT 403 students will complete a final project where they read and implement an advanced algorithm or data structure, CMPUT 303 students do not have a final project
- CMPUT 403 students have a seminar, three times during the year there will be a graded mini-contest where teams of 3 students will work through a problem set.
- CMPUT 403 students and CMPUT 303 students will complete different "open pool" problems (see below).
- CMPUT 403 will be most beneficial to students interested in competitive programming or advanced algorithms.
- CMPUT 303 will be most beneficial to students who are mostly interested in "interview questions" (though a large number of questions will be harder than typical interview questions) or simply honing their problem-solving skills at an intermediate level.

Students in both courses will also gain extensive experience with coding and debugging!

History of the Course: Why a Cross-Listing?

CMPUT 403 was first offered around 2010 and was designed to be a training course in many common tools and algorithm design concepts required to succeed in the International Collegiate Programming Competition (https://icpc.global/). It is a very work-intensive course; students are expected to solve a myriad of difficult problems, some involving advanced topics that may seem a bit niche outside of competitive programming (CP).

Interest in the course grew beyond CP participants and it became clear there is a need to offer a 3rd-year version of the course that was a bit less intensive and did not span quite as many topics. So we offer CMPUT 303 as an alternative to CMPUT 403: students in CMPUT 303 still solve almost the same number of problems as students in CMPUT 403 but the average difficulty is lower and some topics covered in CMPUT 403 are not required to solve CMPUT 303 problems (these topics will be clearly indicated on eClass).

As such, a few algorithmic topics are required only for CMPUT 403 students. But CMPUT 303 students are, of course, welcome to attend these lectures and learn something new even though they are not asked to apply it to any assessments. And students should understand that even CMPUT 303 will still involve considerable effort outside of the meeting times.

Finally, the process of creating this cross-listed course necessitated the addition of a seminar component to CMPUT 403. These seminars largely focus on the development of team work for the purpose of improving performance during an actual competition.

Course topics

Different weeks focus on different categories of problem types and the algorithms required to solve them. Some are more fundamental: dynamic programming, greedy, etc. Some are more focussed: graph algorithms, string algorithms, etc. The following is the weekly breakdown of topics covered in CMPUT 403, each will have an accompanying problem set.

CMPUT 303 students will be required to cover only a subset of these topics. Any topic beginning with an asterisk (*) is not required for any problem issued in CMPUT 303

1. **Introductory problems**
   Emphasis on ad-hoc problems and just getting used to solving problems with the course's online platform.
2. **Built-in algorithms + data structures, binary search**
   Using standard libraries to solve problems. Interesting applications of binary search in problem solving.
3. **Brute force and greedy**
   Trying everything and trying very little - two contrasting approaches.
4. **Dynamic Programming**
   Decomposing a problem into a small number of subproblems - the idea is somewhere between brute force and greedy: you try everything but keep a table of answers for what you have already tried.
5. **Basic Graph Theory**
   Graph searches, spanning trees, (*) strongly connected components.
6. **Combinatorics and Probability**
   Basic counting, binomial coefficients, recurrence relations, (*) probability problems.
7. **String Algorithms**
   KMP matching algorithm, tries, (*) string hashing.
8. **Geometry**
   Primitives (lines, circles, intersections, dot/cross products), convex hulls, Pick's theorem, point-in-polygon, area of a polygon, (*) Euler's theorem (planar graphs).
9. **Number Theory**
   Factoring, sieving for primes, (*) multiplicative functions, (*) quadratic residues, (*) Chinese remainder theorem.
   The problem set for CMPUT 303 students will only include some topics from this week, other problem(s) in this weekly set will involve more practice with dynamic programming.
10. **External Data Structures**
    DIY data structures that are generally not in standard libraries. Binary lifting paradigm and LCA queries, segment trees.
11. **Matchings and Flows**
    Bipartite matching, vertex covers and independent sets in bipartite graphs, (*) maximum flow, (*) max flow / min cut theorem and applications.
    CMPUT 303 students will have some additional basic graph theory problem(s) instead of the flow problems.

12. **(*) Numerical Algorithms**
    Row reduction procedures, linear programming, fast Fourier transform.
13. **(*) O() Optimization Tricks and Randomized Algorithms**
    Some parting notes on little tricks not in textbooks for improving O() running times in some settings. Additionally, how can using random bits sometimes help you write considerably faster programs?

LEARNING RESOURCES

There is no required textbook for this course. But I will be roughly following the sequence of topics given by the book “Competitive Programming 4” by Felix Halem, Steven Halem, and Suhendry Effendy. This is the *recommended* text for the course, see the book’s webpage for more details:
https://cpbook.net/

This “book” is actually split between two books, together they comprise the main reference for this course and students who want to buy this book are encouraged to get both books 1 and 2.

Many students indicate they like this book as a companion for their competitive programming learning. Some students did not like the book and were content with only using my slides and other sources of information they found online about algorithmic topics.

Other (free) supplemental material will be linked to from eClass.

Programming Club Meetings
The U of A programming club meets periodically throughout the week. You can sign up for the mailing list by clicking the appropriate link from the club's page: https://uapspc.github.io/. We don’t really use the mailing list much anymore, but you will be given a Discord invitation when you join the mailing list: most communication happens there.

This is a student-run club: CMPUT 303/403 students are not required to attend. But attendance is encouraged so you can hear about how to solve other challenging programming contest problems that have the same flavour as CMPUT 303/403 problems. Once the meeting schedule for the programming club is decided, I will announce it on eClass.

Again, this is not officially associated with the course in any way. So there is no obligation for the programming club meetings to be compatible with your schedule.

On-Line Homework Disclaimer:

On-line homework is a key component of this course and is provided by a third-party company Kattis via https://ualberta.kattis.com/. The details of your submissions to this site will not be visible to anyone except the instructors and TAs (both present and future) of the course. The only aspect of your submissions that may be visible is a "ranklist" that mentions who has the fastest submissions by programming language among all accepted submissions to any problem on https://ualberta.kattis.com/. This history persists even after the course closes. This is just a byproduct of the website, these rankings have no bearing on the marks you will receive in this course.

An example of what may be seen publicly with your submissions can be found here: https://ualberta.kattis.com/problems/ants/statistics

Students will be required to create an account using their @ualberta.ca email address. There are no additional fees to students for using this platform.

Academic Success Centre:
The Academic Success Centre provides professional academic support to help students strengthen their academic skills and achieve their academic goals. Individual advising, appointments, and group workshops are available year round in the areas of Accessibility, Communication, Learning, and Writing Resources. Modest fees apply for some services.

GRADE EVALUATION

Do not forget to read the Collaboration Policies below!

This section is longer than most other course syllabi because of the likely-unfamiliar nature of these assessments. Read it carefully and bring any questions you might have to the instructor's attention.

CMPUT 303 Students

- Weekly problem sets - 80% split evenly between 11 weekly assignments.
  See "Weekly Assignments" below for further information.
- Open problem pool - 20% for solving a total of 10 problems from a pool of problems.
  Each problem is worth 2% of this category. See "Open Problem Pools" below for more information.

CMPUT 403 Students
As mentioned earlier, CMPUT 403 is a more involved course than CMPUT 303. Students will

roughly solve the same number of weekly problems and open pool problems, but there are two additional categories.

- Weekly problem sets - 60% split evenly between 11 weekly assignments.
  See “Weekly Assignments” below for further information.
- Open problem pool - 15% spread between problems of varying difficulty. See “Open Problem Pools” below for a more detailed breakdown.
- Seminar problems - 10%
  During the seminars on Sep 28, Oct 26, and Nov 23. See “Seminar Contests” below.
- Final project - 15%
  See “Final Project” below for more information.

Final Grades

See the Department of Computing Science’s final grade policies here:
https://www.ualberta.ca/computing-science/resources/policy-information/department-course-policies.html
In particular, it states that we do not necessarily provide grade cutoffs in advance.

But I can give the following guarantees for CMPUT 303/403:

- >= 92% will earn at least A
- >= 88% will earn at least A-
- >= 80% will earn at least B
- >= 65% will earn at least C
- >= 50% will earn at least D (i.e. I will not assign a failing grade to anyone who earns at least 50%)

These guidelines are roughly based on overall student performance for past offerings of the course. The final grades may be assigned a bit differently, if so it will be in the student’s favour (i.e. I will not increase any of the stated cutoffs).

Problem Submissions - How Are They Graded?

Students will submit problems in both the weekly problem sets, for the open problem pool, and (403 only) for the seminar contests. Individual problems are graded the same way in all cases.

- Each submission that is to be counted for marks must have a completed “header” (except for seminar contest problems where a header is not required). A sample header can be found on eClass, you must adapt it for your submission and programming language. In particular, the header should contain:
  - A single line containing `BEGIN-HEADER`
  - Your name and student ID

    - A list of any resources you used to solve the problem. See the **Statement of Expectations for AI Use and Other Online Resources** below for details.
    - A list of any person you discussed the problem with. See the **Collaboration Policies** below for details.
    - The following declaration (which is already in the sample header):
      `By submitting this code, you are agreeing that you have solved the problem in accordance with the course policies in CMPUT 303/403.`
    - A final line containing the text `END-HEADER`

  If your submission does not have a correct header, it cannot be counted for marks: **no exceptions**. It is your responsibility to use the correct header. I recommend the first step in coding a solution is to include the header so you do not forget.
- If one of your submissions receives the verdict **accepted** by the online judge before the deadline and it contains a completed header, you earn 100% for that problem.
- Otherwise, if one of your submissions receives the verdict **accepted** by the online judge before the **COURSE CUTOFF** (see below) but after the deadline for that submission and it contains a completed header, you earn 50% for that problem.
- Otherwise, if none of your submissions receives the verdict **accepted** by the online judge before the **COURSE CUTOFF**, you do not earn any points for that problem.

COURSE CUTOFF

We do accept late submissions for weekly problem sets even as late as the end of the course, this is to ensure you have incentive to finish a problem you had worked on but may not have been able to finish on time. The instructor will be more willing to give larger hints towards solving problems when their deadlines have already passed.

That being said, we do have to cut off submissions completely by some time. Friday, Dec 8, 2023 at 11:55pm is the **COURSE CUTOFF**, any submissions received after that time will not be eligible for marks. This applies to all work in the course: the weekly problem sets, the open problem pools, and the final project (CMPUT 403 only).

Weekly Assignments

Nearly every week, a problem set will be posted to eClass, this list will clearly indicate which problems are assigned to CMPUT 303 students and which problems are assigned to CMPUT 403 students.

In each weekly problem set, all problems have the same weight regardless of their difficulty. For example, the first week has 7 problems. If you earn **accepted** for 5 of these problems before the assignment deadline, and earned **accepted** for one of these problems before the **COURSE CUTOFF**, and never solved one of them then you earned 5.5 out of 7 marks for that assignment

which is around 79.6% (again, this is assuming you included a completed header for your submission).

The last two weeks of topics do not have associated weekly exercises as students will undoubtedly be busy finishing the open pools and, for CMPUT 403, the final project. Some problems in the open pools can be solved with these topics for those that want to try them out.

Open Problem Pools
Both CMPUT 303 and CMPUT 403 must solve problems from problem pools, but they differ between 303 and 403. What distinguishes the open problem pool from the weekly problem sets is that the problems are presented without context: you are not immediately told which topics you need to draw from in order to solve the problems. They are due at the **COURSE CUTOFF**.

CMPUT 303 students will solve 10 problems from their open problem pool which will consist of roughly 20 to 25 problems. Each submission is worth 2% of the 20% total for the open problem pool for CMPUT 403. These are problems selected by the instructor and TAs that can be solved by applying ideas from the topics required for CMPUT 303 students in this course.

The open problem pool in CMPUT 403 is slightly more involved: There will be 3 pools of problems: easy, medium, and hard. Here is how it is graded (percentages are out of the 20% total for the open problem pool in CMPUT 403):

- An easy problem is worth 1%.
- A medium problem is worth 2%.
- A hard problem is worth 3%.
- You are capped at 2% in the easy problem category and, after capping the easy problems, you are capped at 12% for easy+medium.
- Of course, you can earn at most 15% overall in this open pool category.
- The precise formula is this. Let E, M, H be the number of easy, medium, and hard problems you solve (with correct headers). Your score for the open pool will be: min(min(min(E,2) + 2*M, 12) + 3*H, 15)

Example: the "easiest" way to get full credit for the open pool would be to solve 2 easy problems, 5 medium problems, and 1 hard problem.

Another example: if you solved 4 easy problems and 4 medium problems but no hard problems, your score would be 10% out of 15% for the open pool: two of the easy problems you solved would not count because the score is capped at 2% for the easy problem category.

Another example: if you just solved 5 hard problems your score would be a full 15% for the open pool.

Another example: if you solved 6 medium problems and 1 hard problem, your score would still be a full 15% for the open pool.

One last example: if you solved 2 easy problems and 10 medium problems, your score would be 12%, only 5 of the medium problems counted because of the cap on easy+medium.

Seminar Contests:
During the seminars on Sep 28, Oct 26, and Nov 23, teams of 2 or 3 students will be given a small set of problems and must solve as many as possible during this 50-minute period. Each contest will be worth 5% of the overall grade in the course and each student's final grade in this seminar portion will be based on their best 2 contests. These three seminars are the only ones with graded activities and students must appear in person to be counted.

For each contest, 2% will be given by default for participation. Each problem solved up to a maximum of 3 will be worth an additional 1%. In particular, your grade does not depend on how well you do relative to other teams in the course. We want you to have a bit of contest experience, but we do not want it to be competitive.

In general, most seminar contest problems will be easier than a typical weekly problem set. Other seminars will focus on discussing good team work strategies, running mock contests to simulate a graded contest, discuss solutions to practice problems in the seminars, get direct feedback and advice from instructors/TAs on real-time problem solving, etc.

Contest rules:
Only one laptop can be used per team, like in an official ICPC contest. You may use any online resource in the same way as you can with weekly problem sets and you can bring as much printed material as you would like (books, printouts, notes, etc.). More guidance on what may be helpful will be given in the seminars prior to the ones with contests. You may not speak with any other team during the contest, only the instructor and TAs. No electronic devices of any kind should be used apart from the one laptop per team. Teams should ensure the laptop can connect to the University's network, has a web browser that works with the Kattis site used in the course, and has appropriate development tools (IDEs, compilers, etc.) for team members to be able to code solutions in the Kattis-supported language of their choice. The participation marks can only be earned for teams that participate on site in the seminar room.

The contest will start immediately at 1:00pm and end exactly at 1:50pm. No late submissions will be accepted. No headers are required for contest submissions. Teams must "register" for the contests on Kattis (instructions will be provided). Problems may be discussed freely after the contest.

We expect there will be enough laptops between the students to ensure each team has access to one. If there are difficulties with this, we will look into an appropriate solution.

Final Project (CMPUT 403 Only):
You will provide an efficient implementation of an algorithm or data structure that is not "standard" in the contest setting but still falls under the general algorithms umbrella (i.e. one that could be taught in an "algorithms" course in either an upper-level undergraduate or graduate course). The exact topic is chosen by you, although it must conform to the guidelines that will be specified in the final project description that will be posted to eClass. For a preview of the sorts of projects that are permitted, see the lists below.

Examples of acceptable topics include, but are certainly not limited to, the following:
`faster heaps, link-and-cut data structures, minimum-cost branchings, randomized algorithms and/or data structures, network simplex, finding 2-player Nash equilibria, faster matching or network flow algorithms, integer factorization/primality testing methods, planarity testing`

Examples of projects that are not in line with course:
`classification algorithms from machine learning, sub-optimal "heuristics", anything already in the code archives, image processing algorithms`

A project proposal will be due by Friday, Nov 3 at 11:55pm and will receive feedback within 3 days of submitting the proposal: earlier submissions are permitted and will receive earlier feedback. The proposal is not for marks, it is just to ensure your thoughts are aligned with instructor expectations on topic and difficulty. Students who fail to submit a proposal do so at their own risk, and the instructor will probably bug them with an email or two!

A full implementation of the project will then be due by the **COURSE CUTOFF** (see above). A more detailed list of deliverables will be included in the final project specification on eClass.

Collaboration Policies:
For definitions of the policies cited below, consult our department's page.
https://www.ualberta.ca/computing-science/resources/policy-information/department-course-policies.html

Collaboration Policy for Weekly Problem Sets and Open Problem Pools.
We are adopting the "consultation model" meaning you may have verbal conversations with classmates on how to solve a problem, but no notes should be made during these conversations.

Do not share code, written ideas (pseudocodes, formulas, etc.), screenshots, etc of any part of a solution whether it is the actual code or just part of conceptualizing ideas. The only exception to this is that you are permitted to share test cases (sample input/output for a problem) that you found helpful.

Do not consult with students outside of the course, consultation should only be between classmates. Consultations between students in CMPUT 303 and CMPUT 403 are permitted. Mention all students you consulted with in the header for your submissions. If you have any questions about what constitutes appropriate collaboration, you should contact the instructor.

Collaboration Policy for the Seminar Contests (CMPUT 403 only)
We are adopting the "teamwork model" for the seminar contests. You may speak freely with your team and share ideas fully both verbally and in written form, but you cannot speak with any other person about the problems during the contest apart from the instructor and TAs. Problems can be freely discussed after the seminar.

Collaboration Policy for the Final Project (CMPUT 403 only)
We are adopting the "solo effort model" for the final project. You should complete it entirely yourself and have no consultation with any other person apart from the instructor and TAs.

Statement of Expectations for AI Use and Other Online Resources:
One of the primary purposes of this course is to gain experience solving problems on your own. You are permitted to use previously-implemented versions of standard algorithms to be used as a tool for solving a problem (eg. using an implementation of Dijkstra's algorithm). Pointers to implementations of standard data structures / algorithms in both C++ and Python will be provided on eClass. If you use such an implementation, cite the source (e.g. provide the URL or some other reference to the implementation).

That being said, the use of previous implementation is not meant to replace the ***problem solving aspect***. Beyond the first week of problems, most problems seem to be unsolvable by ChatGPT 3.5. That being said, one may still be tempted to use a tool like this to help solve a given problem (eg. maybe a solution generated by ChatGPT or Github Copilot is "close" and is easy to debug). **This undermines the intent of the course**. As such, the use of ChatGPT, Github Copilot, etc. is strictly forbidden.

Here is a simple criteria for you to follow when understanding which resources can be used:

- Is the resource created specifically to solve an assignment problem or a very similar problem (i.e. the problem solving will be done by someone else, not you)? **Forbidden!**

- Is the resource an implementation of a generic algorithm that is not used specifically to solve the problem (e.g. Dijkstra's algorithm, line intersection algorithm, segment tree data structure, etc.). **Permitted with citation.**
- Is the resource one that is *generated* after you enter some information (eg. ChatGPT, posing a Stack Exchange question, Github Copilot suggestion, etc.). **Forbidden!**
- Is the resource just generic questions about a programming language (e.g. looking up how C++ range-based for loops work) and has nothing to do specifically with the problem at hand? **Permitted with citation.**

Contact the instructor if you have any questions about the permissibility of a particular resource.

Re-evaluation of Term Work:
There is nothing to "re-evaluate" for the weekly assignments and the open pools during the term since all problems are pass/fail problems as determined by the online judge. We will issue a grading script that you can use to see a summary of your progress in the course so far. After the last problem set is due, we will prepare and post a final summary of your performance in the weekly problem sets and open pools. You will have 3 days to inform us if there is a discrepancy.

POLICIES FOR LATE AND MISSED WORK

Late Policies:
As discussed above, any problem from a weekly problem set that is "accepted" by the judge after the problem set's deadline but before the final **COURSE CUTOFF** as defined above will receive 50% credit. Any weekly problem set submitted after the **COURSE CUTOFF** will not receive any credit. Since the course cutoff is when the open problem pools are due, we cannot accept late submissions for the open problem pools. Similarly, the final projects (CMPUT 403) cannot be submitted late - students who are unable to complete the projects in time should submit what they have at the deadline and are encouraged to document what they completed and what they did not complete to make it easier to assess their submission. Seminar contest problems are due by the end of the seminar that day at 1:50pm, no late submissions will be graded.

Missed Term Work or Final Exam Due to Non-medical Protected Grounds (e.g., religious beliefs):
When a term assessment or final exam presents a conflict based on non-medical protected grounds, students must apply to the Academic Success Centre for accommodations via their Register for Accommodations website. Students can review their eligibility and choose the application process specific for *Accommodations Based on Non-medical Protected Grounds*.

It is imperative that students review the dates of all course assessments upon receipt of the course syllabus, and apply ***AS SOON AS POSSIBLE*** to ensure the timely application of the accommodation. Students who apply later in the term may experience unavoidable delays in the processing of the application, which can affect the accommodation.

Missed Term Work for Weekly Problem Sets:
A student who cannot complete a weekly problem set due to incapacitating illness, severe domestic affliction or other compelling reasons must contact the instructor within two working days of missing the assessment, or as soon as possible, to request an excused absence. If an excused absence is granted, then the deadline for that submission may either be extended or the weekly assignment category weight will be redistributed over the completed weekly exercises. An excused absence is a privilege and not a right. There is no guarantee that an absence will be excused. Misrepresentation of facts to gain an excused absence is a serious breach of the Code of Student Behaviour. In all cases, instructors may request adequate documentation to substantiate the reason for the absence at their discretion.

STUDENT RESPONSIBILITIES

Academic Integrity and Student Conduct:
The University of Alberta is committed to the highest standards of academic integrity and honesty, as well as maintaining a learning environment that fosters the safety, security, and the inherent dignity of each member of the community, ensuring students conduct themselves accordingly. Students are expected to be familiar with the standards of academic honesty and appropriate student conduct, and to uphold the policies of the University in this respect. Students are particularly urged to familiarize themselves with the provisions of the Code of Student Behaviour and the Student Conduct Policy, and avoid any behaviour that could potentially result in suspicions of academic misconduct (e.g., cheating, plagiarism, misrepresentation of facts) and non-academic misconduct (e.g., discrimination, harassment, physical assault). Academic and non-academic misconduct are taken very seriously and can result in suspension or expulsion from the University.

All students are expected to consult the Academic Integrity website for clarification on the various academic offences. All forms of academic dishonesty are unacceptable at the University. Any suspected academic offence in this course will be reported to the College of Natural and Applied Sciences. Suspected cases of non-academic misconduct will be reported to the Dean of Students. The College, the Faculty of Science, and the Dean of Students are committed to student rights and responsibilities, and adhere to due process and administrative fairness, as outlined in the Code of Student Behaviour and the Student Conduct Policy. Anyone

who is found in violation is likely to receive a sanction. Typical sanctions for academic misconduct include conduct probation, a mark reduction or a mark of 0 on an assessment, a grade reduction or a grade of F in a course, a remark on the transcript, and a recommendation for suspension or expulsion. Sanctions for non-academic misconduct include conduct conditions, fines, suspension of essential or non-essential University services and resources, and suspension or expulsion from the University.

Cell Phones:
Cell phones are to be turned off during lectures, labs and seminars.

Students Eligible for Accessibility-Related Accommodations:
In accordance with the University of Alberta's Discrimination, Harassment, and Duty to Accommodate policy, accommodation support is available to eligible students who encounter limitations or restrictions to their ability to perform the daily activities necessary to pursue studies at a post-secondary level due to medical conditions and/or non-medical protected grounds. Accommodations are coordinated through the Academic Success Centre, and students can learn more about eligibility on the Register for Accommodations website.

It is recommended that students apply as early as possible in order to ensure sufficient time to complete accommodation registration and coordination. Students are advised to review and adhere to published deadlines for accommodation approval and for specific accommodation requests (e.g., exam registration submission deadlines). Students who request accommodations less than a month in advance of the academic term for which they require accommodations may experience unavoidable delays or consequences in their academic programs, and may need to consider alternative academic schedules.

Recording and/or Distribution of Course Materials:
Audio or video recording, digital or otherwise, of lectures, labs, seminars or any other teaching environment by students is allowed only with the prior written consent of the instructor or as a part of an approved accommodation plan. Student or instructor content, digital or otherwise, created and/or used within the context of the course is to be used solely for personal study, and is not to be used or distributed for any other purpose without prior written consent from the content author(s).

Learning and Working Environment:
The Faculty of Science is committed to ensuring that all students, faculty and staff are able to work and study in an environment that is safe and free from discrimination, harassment, and violence of any kind. It does not tolerate behaviour that undermines that environment. This includes virtual environments and platforms.

If you are experiencing harassment, discrimination, fraud, theft or any other issue and would like to get confidential advice, please contact any of these campus services:

- Office of Safe Disclosure & Human Rights*: A safe, neutral and confidential space to disclose concerns about how the University of Alberta policies, procedures or ethical standards are being applied. They provide strategic advice and referral on matters such as discrimination, harassment, duty to accommodate and wrong-doings. Disclosures can be made in person or online using the Online Reporting Tool*.
- University of Alberta Protective Services: *Peace officers dedicated to ensuring the safety and security of U of A campuses and community. Staff or students can contact UAPS to make a report if they feel unsafe, threatened, or targeted on campus or by another member of the university community.*
- Office of the Student Ombuds*: A confidential and free service that strives to ensure that university processes related to students operate as fairly as possible. They offer information, advice, and support to students, faculty, and staff as they deal with academic, discipline, interpersonal, and financial issues related to student programs.*
- Office of the Dean of Students*: They can assist students in navigating services to ensure they receive appropriate and timely resources. For students who are unsure of the support they may need, are concerned about how to access services on campus, or feel like they may need interim support while you wait to access a service, the Dean of Students office is here to help.*

Feeling Stressed, Anxious, or Upset?
It's normal for us to have different mental health experiences throughout the year. Know that there are people who want to help. You can reach out to your friends and access a variety of supports available on and off campus at the Need Help Now webpage or by calling the 24-hour Distress Line: 780-482-4357 (HELP).

Student Self-Care Guide:
This Self-Care Guide, originally designed by the Faculty of Native Studies, has broader application for use during students' learning. It provides some ideas and strategies to consider that can help navigate emotionally challenging or triggering material.

Policy about course outlines can be found in Course Requirements, Evaluations Procedures and Grading of the University Calendar.

Land Acknowledgement:

The University of Alberta respectfully acknowledges that we are situated on Treaty 6 territory, traditional lands of First Nations and Métis people.

To learn more about the significance of this land acknowledgement, please read this useful article and associated links to more information.

Disclaimer:

Any typographical errors in this Course Outline are subject to change and will be announced in class. The date of the final examination is set by the Registrar and takes precedence over the final examination date reported in this syllabus.