**COMP 1805: Discrete Structures I**
**Fall 2021**
**Syllabus**

Introduction to discrete mathematics and discrete structures. Topics include: propositional logic, predicate calculus, set theory, complexity of algorithms, mathematical reasoning and proof techniques, recurrences, induction, finite automata, and graph theory. Material is illustrated through examples from computing. **Precludes** additional credit for MATH 1800. **Prerequisite(s)**: one Grade 12 university preparation mathematics course. **Minimum grade of C-** in COMP 1805 is required in order to take COMP 2804, COMP 3005, COMP 3007, or COMP 4001.

| | |
|---|---|
| **Instructor** | Alexa Sharp (she/her) --- call her Alexa |
| **Email** | **alexasharp3@cunet.carleton.ca** |
| **Lectures** | if you prefer live: Wed & Fri 2:35pm - 3:55pm |
| | if you prefer asynchronous: recordings provided |
| **Tutorials** | if you prefer live: Wed 10:35am - 11:25am **or** Mon 2:35pm - 3:35pm |
| | if you prefer asynchronous: recordings provided |
| **Office Hours** | schedule held on discord voice channel |
| **Course Resources** | lecture for most up-to-date zoom id, see brightspace<br>piazza piazza.com/carleton.ca/fall2021/comp1805<br>schedule topics, readings, slides, videos, deadlines<br>brightspace https://brightspace.carleton.ca/d2l/home/<br>gradescope https://www.gradescope.ca/courses/4507<br>textbook https://cs.carleton.edu/faculty/dln/book/ |
| **Q&A Forums** | piazza (primary, structured), discord (non-anonymous, informal) |

## tl;dr

This course is meant to introduce you to the language of discrete mathematics. As with any language, you have to learn a lot of vocabulary and rules in order to speak the language, and then you have to practice, practice, practice. The lectures provide the vocabulary and rules, and the drills, tutorials, and problem sets provide the opportunity and incentive to get this practice with varying degrees of guidance. Every aspect of the course, every question, every evaluation, serves a purpose to get you to our final goal of you being able to fluently speak (discrete) mathematics, so that you can be a kick-ass problem solver. But you must put in the practice.

In a **perfect** world, your course workflow would be:

- Before lecture, *skim* the reading for that lecture in the textbook (5-15 mins/lecture).
- Attend or watch each lecture to be exposed to the new material (80 mins/lecture).
- After lecture, do online drills to practice the vocabulary, notation, and high-level concepts (e.g. what does that word or symbol mean again? When is this approach better than the previous one?) (10-30 mins/lecture).
- Attempt the tutorial problems, watch and/or attend tutorial, then complete accompanying online exercises to get hands-on practice with some guidance (e.g. make one sentence at a time using new vocabulary and rules) (2-3 h/week).
- Complete each problem set to get more independent practice (e.g. write an essay using your new vocabulary and new ideas), reflect upon one's improvements in the self-evaluation (4-15h each).
- Reference the textbook and lecture slides as needed.

Of course, this workflow won't work for you all, but it is something to aspire to.

## Lectures

**Whatever your course delivery preference, there is an option for you.** Regardless of section, you may view lectures completely asynchronously if you choose, with the option to attend lectures synchronously (live) during section B's time, if your schedule permits. Any synchronous lectures will be on Zoom, with recordings and transcripts provided within 12 hours (but usually within 2). You can find the most up-to-date zoom lecture link on piazza; please do not share this link with anyone outside of this course. In live lectures, please mute your microphone, but turn on video if your internet connection and living arrangements can handle it. If you have questions, please ask by typing it in the chat; Alexa will get to it when she can. Do not use the chat for irrelevant comments so that Alexa has a chance to see your questions. If you wish not to be recorded, simply leave your camera and microphone turned off.

## Tutorials

Regardless of section, you may view tutorials asynchronously if you choose, with the option to attend synchronously (live) if your schedule permits (during section B1 or B2's time). Any live tutorials will be on Zoom, with recordings and transcripts provided within 12 hours.

## Textbook

We will use David Liben-Nowell's *Connecting Discrete Mathematics and Computer Science*. A free PDF version of the book is available at https://cs.carleton.edu/faculty/dln/book/.

## Course Work & Evaluation

### Important Dates & Deadlines

| | |
|---|---|
| **Drill Practice** | 2:30pm of the Friday the week after associated lecture |
| **Tutorial Practice** | 2:30pm of the Friday the week of associated tutorial |
| **Problem Set 1** | 2:30pm on **Monday Sep 27** |
| **Problem Set 2** | 2:30pm on **Monday Oct 11** |
| **Test #1 part 1 (brightspace)** | 60 mins (your choice) **Oct 11 3:25pm - Oct 15 2:30pm** |
| **Test #1 part 2 (take-home)** | **Friday Oct 15 4:00pm - Monday Oct 18 2:30pm** |
| **Problem Set 3** | 2:30pm on **Monday Nov 8** |
| **Problem Set 4** | 2:30pm on **Monday Nov 22** |
| **Test #2 part 1 (brightspace)** | 60 mins (your choice) **Nov 22 3:25pm - Nov 26 2:30pm** |
| **Test #2 part 2 (take-home)** | **Friday Nov 26 4:00pm - Monday Nov 29 2:30pm** |
| **Problem Set 5** | 2:30pm on **Friday Dec 10** |
| **Self-Evaluation** | 11:55pm on **Friday Dec 10** |
| **Final Test** | TBD |

### Grade Computation

| | |
|---|---|
| Drills | 8% |
| Tutorials | 6% |
| Problem Sets | 48% |
| Tests | 16% |
| Final | 16% |
| Self-Evaluation | 6% |

If you want to compute your grade yourself, you can make a copy of this spreadsheet and fill in your grades as you go along to know where you stand.

### Schedule, Readings, Lecture Notes & Recordings

Please refer to the detailed course schedule for suggested textbook readings, lecture slides (pre- and post-markup), lecture and tutorial recordings, as well as links to problem sets, drills, tutorials, and any other coursework. You'll want to bookmark the page.

**Drills (8%)**

While the end goal of this course is to get you comfortable writing your own full-blown proofs, there are a lot of necessary "building block" skills to pick up first. Just as a soccer player prepares for matches with passing and shooting drills, you will prepare for your proof-based problem sets with focused drills that target vocabulary, notation, and approaches you need to solve problems and communicate your solution. The idea is to make the "smaller stuff" second nature so that you can more easily put it together into masterful, comprehensive proofs. This is what the drills aim to provide.

After each lecture, I will post 1-10 lecture-related "drill" questions. These are meant to take a few minutes each, and provide technical practice with new definitions and notations, or hone your intuition for the material. They will often focus on the big picture--on the ability to compare and contrast, and to make quick assessments based on intuition--which should help give you a better sense of direction when attacking a new problem set. Also, they are similar to short-form test questions---in fact, ~10% of each test and ~20% of the final are drawn directly from the drills (with only very minor adjustments).

**There are ~25 drills (~1 per lecture). To account for emergencies, the best 16 are worth .5% each;** the remaining ~9 are dropped (or, more specifically, used as bonus). Drills (on brightspace) are due by 2:30pm on Friday of the week following the associated lecture. Multiple attempts within the week are allowed.

**Tutorials (6%)**

While the drills provide specific, fine-tuned practice, they will not involve writing proofs. The tutorials guide you through solving a new problem, and ask you to practice on your own with low-stakes evaluation. As with the drills, ~10% of each test and ~20% of the final are drawn directly from the tutorial exercises (with only minor adjustments).

While each tutorial involves working through 2-3 proofs-based problems, you are evaluated on the multiple-choice style brightspace questions (which are based on the 2-3 problems).

**There are ~10 Tutorials (~1 per week, 2 used for tests). To account for emergencies, the best 6 are worth 1% each;** the remaining ~4 tutorial grades are dropped (or, more specifically, used as bonus.) Tutorial videos will be 20-40 minutes long and will be available prior to Tutorials. You may attend a tutorial live during section B1 and B2 (regardless of your section) to work through the problems with the guidance of a TA. Tutorial exercises (on brightspace) are due by 2:30pm Friday of the week of the associated tutorial.

**Problem Sets (48%)**

The best computer scientists are the ones who have had the most practice. The problem sets in this course are meant to give you the opportunity to practice the topics of this course in a way that is challenging yet also manageable. **You should be out of your comfort zone** (but not overwhelmed.) At times you may struggle and at others it may seem more straight-forward; just remember to keep trying and practicing, and over time you will improve. Everyone learns differently; be patient with yourself and you will improve. The only way to really practice with the problem solving process is to experience it.

While you are encouraged to collaborate with your peers, you should write up your solutions on your own. You may not use web search/the internet to look for approaches to the problem set; if you need help, we have many ways to do so without violating academic integrity.

Note that 10% of each problem set is set aside for an optional self-evaluation. See the self-evaluation section below for more information.

**There are 5 problem sets. To account for emergencies, the best 4 are worth 12% each;** the lowest problem set grade will be dropped (or, more specifically, used as bonus.) **Please "save" your dropped problem set for unforeseen emergencies**. I cannot drop 2 problem sets.

Typeset problem set solutions are due at 2:30pm on Mondays on gradescope. **Scanned, photographed or non-pdf documents will receive 0 marks.** Lates are accepted within 24 hours, after which no lates are accepted due to posted solutions.

**Tests & Final (8%, 8%, 16%)**

While tests aren't particularly representative of how you may use your computer science knowledge in practice, they provide some advantages:

- Studying for the tests hopefully improves your memory so that some of the more fundamental information can be recalled quickly.
- Tests encourage you to review all the course material, not just what is needed for the problem sets, tutorials, and drills.
- Tests are the only coursework that evaluates you as an individual, since the remaining coursework can be collaborative to some extent.

Having said that, both the tests and the final are worth around as much as a problem set.

**The tests are worth 8% each and the final is worth 18%.** The tests have 2 parts: the first part consists of multiple-choice-style questions that must be answered on brightspace during a 60-minute time-bounded period of your choice during the week in question. The second part consists of free-form questions that you will have ~3 days to work on, type up, and hand in on gradescope. The final will be multiple-choice-style questions that must be answered on

brightspace during our scheduled final time (TBD). The final is cumulative. More information about the tests will follow.

20% of the tests and 40% of the final will be drawn from the drill and tutorial exercises, so you will know a bit what to expect. You do not need to pass the tests or final to pass the course.

**Self-Evaluation (a.k.a. Show Me Your Intangibles) (6%)**

Disclaimer: this is a pedagogical experiment. I will always compute two grades for you: one with the experiment and one without, then I will assign you whichever is best. **The self-evaluations are meant to alleviate stress and anxiety, so if it causes you any, please just ignore them.**

Problem sets and tests generally evaluate you based on the extent to which you are able to complete assigned problems. When you are not able to complete a problem, your grade does not always reflect the partial learning you have achieved, or the skills that you have improved in attempting the problem. The self-evaluation is meant to be a place for you to reflect on what you have learned and what skills you have improved in attempting the assessment in question. It is a way for you to indicate your engagement with the material, even if you weren't able to "get" everything.

**10% of each problem set, and 6% of your course grade can be based on self-evaluations.**

To help you with your self-evaluation, I will provide a list of possible skills to consider in each evaluation. You will be asked to choose a letter grade for your evaluation and write a few short sentences of justification. This grade may be higher or lower than what you end up getting on the remaining 90% of your problem set or 94% of your course grade. If you self-evaluate lower than your remaining grade, you will get your maximum grade. (If you self-evaluate higher, you still get the higher combined grade.) In this way, you are not ever penalized by participating in the experiment.

Note that I am not asking you to guess your problem set grade; rather, this is a way for you to adjust your problem set grade to better reflect the learning and practice you have achieved. I am trying this out because I have observed that often a student's grade does not reflect some of the intangible skills I witness them learning. Please try to be truthful; if everyone always gives themselves the top grade and there doesn't appear to be great self-reflection (it's only a few sentences folks!) then I will no longer use the experiment in subsequent semesters.

What am I hoping to get out of this? Firstly, I hope this gives you some sense of agency in your assessment. Secondly, I hope your course grades better reflect the learning and practice you have done throughout the course. Finally, I hope to more closely examine the discrepancies between self-assessments and non-self-assessment grades, to see if I can better design future coursework and assessments to narrow any gaps.

But remember: if you just give yourself 0% on all self-assessments, you will still get the same course grade as if we didn't have any self-assessments at all. So feel free to ignore them at any point, but I hope that you don't :-)

**Late Policy & Emergencies**

Drill and tutorial exercises must be submitted by the deadline, no lates accepted.

Problem set solutions may be handed in up to 24 hours late without question, after which no lates are accepted, no exceptions.

In the case of an emergency, you can, without explanation, not complete up to ~9 drills, ~6 tutorials, and 1 problem set. If you need more accommodations, please contact the prof but be aware that dropping more marks is likely not possible. You need to do a minimum amount of proofs practice in this course in order to deeply engage with the material. For this reason, do not "squander" the emergencies for non-emergencies.

**Proofs**

This course is about learning enough discrete mathematics to be able to read, write, appreciate, and find the logic flaws in a variety of mathematical proofs. As such, you will see that over the course of the semester, at least 50% of the marks overall will be for your mathematical proofs. Writing your own proofs will likely feel awkward and even overwhelming, as any new endeavour can feel. I encourage you to acknowledge the struggle, ask for help, and just put in the practice. With practice you will improve, but you cannot improve without practice.

**Bonus (a.k.a. Extra Credit)**

There are many opportunities for bonus points, if you have the time and inclination. Any remaining problem sets, tutorials, and drills outside of your best submissions (as described above) are available for bonus.

**Bonus points are completely optional**; *not* doing bonus points will not negatively impact your final grade. Bonus points cannot add more than 3% to your final mark, and it cannot move you from an F to a passing grade.

## Learning Outcomes

By engaging with the course material through practice, a student should:

1. become comfortable reading and using mathematical terminology (i.e. vocabulary)
    - sets, functions, propositional and predicate logic, asymptotic notation, recursion, graphs

2. become comfortable reading and writing beautiful mathematical proofs
    - direct proofs, proofs by counterexample, proofs by construction, proofs by contradiction, proofs by contrapositive, proofs by induction
3. improve problem solving and critical thinking skills, such as
    - using examples, counter-examples, diagrams, simpler cases, similar problems, etc., to better understand a mathematical statement,
    - recognizing a broken proof or a false start and using them to find a new result or approach,
    - thinking critically about which proof paradigm is most appropriate.

## Necessary Equipment, Accounts & Software

You will need an internet-connected device to access lectures and slides, to access brightspace for drills, and tests, gradescope for problem sets, and piazza and discord for Q&A and office hours.

You will need (free) accounts on piazza, gradescope, and discord; please use your preferred (recognizable) name on all of these platforms. You may optionally want (free) accounts on overleaf and anki for LaTeX typesetting and flashcards, respectively.

You don't need any special software for this course, but you need a way of typesetting mathematical symbols and producing a **pdf** document (Google Docs, Microsoft Office, or LaTeX can all do this.) **A scanned, photographed or non-pdf document will receive 0 marks.**

## How to Get Help

There are many ways to get help on your work in this course that do not violate the course's academic integrity policy:

- Reference the relevant proof paradigm sheets (posted on piazza)
- Reference the problem solving tips sheet (posted on piazza)
- Check the Q&A Forum on piazza and ask questions there
    - please read the posting etiquette so that you get the fastest, most useful answer
- Go to Office Hours on discord (schedule posted on piazza and on this spreadsheet)
    - good for questions that require more back-and-forth, that cannot be answered on piazza or discord
- Check the Q&A on discord (ideally after you've searched piazza to confirm it hasn't been answered there.)

## Academic Integrity

You **may** talk with peers and TAs at a high-level. But **you must write up your solutions on your own**. If your solutions are basically word-for-word the same as a peer's, that's too close.

You **must not** show or otherwise share your solution with your peers or on the internet.

You **must not** use the internet to search for or solicit approaches or ideas.

You **must not** post our problem sets or solutions on the internet, before or after the due date.

Any student that violates academic integrity (intentionally or not) must be reported to the Associate Dean (Undergraduate) who will investigate the matter. The standard penalties for an academic integrity violation are as follows:

- First offence: F in the course.
- Second offence: One-year suspension from the program.
- Third offence: Expulsion from the University.

These are standard penalties. More-severe penalties will be applied in cases of egregious offences. For more information, please see Carleton University's Academic Integrity Policy.

## Respect in the Classroom and Forums

Please remember to treat your peers and the course staff with respect. This includes in the zoom chat and on any course-related forums such as piazza and discord. It is not acceptable to use offensive language nor disparage a person or group, no matter the intent. Treat the course spaces as professional spaces and behave accordingly. Behavioural misconduct may be reported to Student Affairs.

We recommend you read over our piazza posting etiquette as well as our discord #rules-please-read channel. You are responsible for behaving within these parameters.

If you feel you have been disrespected or abused either by other students or course staff, please let us know (you can **contact us anonymously and privately on piazza**, for example.)

# Statement of Accommodation

The Carleton University Information on Academic Accommodation applies to this course. Here is information on how to apply for academic accommodation. If you are allowed extra time on tests, you will get an email at least a week before the test to coordinate. If there is anything Alexa can do to help you succeed, please let her know as soon as possible so that she can accommodate accordingly.

## Late Policy

Please refer to the Late Policy & Emergencies section above.

## Copyright

Lectures and course materials (including all slides, programs, handouts, videos, and similar materials) are protected by copyright. Alexa is the exclusive owner of copyright and intellectual property of all course materials. You may take notes and make copies of course materials for your own educational use. You may not reproduce or distribute lecture notes and course materials publicly for commercial purposes, or allow others to, without express written consent.

## Territory Acknowledgement

I would like to acknowledge that the location of the Carleton University campus is on the traditional, unceded territories of the Algonquin nation. In doing so, I acknowledge that I and Carleton University have a responsibility to the Algonquin people and a responsibility to adhere to Algonquin cultural protocols.

More information about how Alexa is trying to take some responsibility can be found on piazza in the post titled "Beyond the Land Acknowledgement."

## Undergraduate Academic Advisor

The Undergraduate Advisor for the School of Computer Science is available in Room 5302C HP, or by email at **scs.ug.advisor@carleton.ca**. The undergraduate advisor can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisor will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

## University Policies

In addition to anything included here, all the standard Carleton University Policies regarding equity and academic regulations apply to this course.

## SCS Tech Support

Technical support information can be found at: https://carleton.ca/scs/technical-support/. Technical support is available in room HP5161 Monday to Friday from 9:00 until 17:00 or by emailing **SCS.Tech.Support@cunet.carleton.ca**.