COMP 2402: Abstract Data Types and Algorithms
Winter 2022
Syllabus

This course builds upon the principles introduced in COMP 1405 and COMP 1406 and provides a general background for further study in Computer Science. The course will cover object-oriented programming concepts; the design and implementation of data structures (linked lists, stacks, queues, trees, heaps, hash tables, and graphs) and related algorithmic techniques (searching, sorting, recursion); and algorithm analysis. Students will be expected to complete a number of programming projects using the concepts presented. **Precludes** additional credit for COMP 2002 (no longer offered), SYSC 2002 (no longer offered), SYSC 2100. Prerequisites: one of COMP 1406, COMP 1006, SYSC 2004, with a **minimum grade of C-**.

Instructor	Alexa Sharp (she/her) --- call her Alexa or Prof Alexa
Email	**alexasharp3@cunet.carleton.ca**
Lectures	if you prefer live: Wed & Fri 2:35pm - 3:55pm if you prefer asynchronous: recordings provided
Student Hours	schedule held on discord voice channel
Course Resources	lecture for most up-to-date zoom id, see brightspace piazza piazza.com/carleton.ca/winter2022/comp2402 schedule topics, readings, slides, videos, deadlines brightspace https://brightspace.carleton.ca/d2l/home/ textbook http://opendatastructures.org/
Q&A Forums	piazza (primary, structured), discord (non-anonymous, informal)

Lectures

There are both synchronous (B) and asynchronous (A) sections of this course, but you can choose your preferred course delivery, schedule permitting. 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. Any synchronous lectures will be on Zoom, with recordings and transcripts provided within 6 hours (but usually within 2). You can find the most up-to-date zoom lecture link on brightspace; please do not share this link with anyone outside of this course.

To reiterate: for Section B, lectures are offered synchronously (live), but live attendance is not mandatory; recordings are provided for later viewing, and Section B students are able to watch all lectures asynchronously if they wish. For Section A, asynchronous recordings are provided on Wednesday and Friday evenings. Section A students are welcome to attend the live Section B lectures if timing permits. The only way in which the two sections are different is the precise time of your midterm test; otherwise you have the same options available to you all.

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; Prof Alexa will get to it when she can. Do not use the chat for irrelevant comments so that Prof Alexa has a chance to see your questions; this includes on discord. If you wish not to be recorded, simply leave your camera and microphone turned off.

Students are expected to remain up to date with the deadlines and due dates provided by Prof Alexa.

Please note that recordings are protected by copyright. The recordings are for your own educational use, but you are not permitted to publish to third party sites, such as social media sites and course materials sites.

Textbook

The textbook for the course is Pat Morin's Open Data Structures. Free PDF and HTML versions of the book are available at opendatastructures.org. You'll probably want to use the Java version. There are also links there explaining where you can purchase a paperback version.

Necessary Equipment and Software

You will need an internet-connected device to access various course materials. You will need a java compiler and your favourite editor to complete the programming assignments.

You will need (free) accounts on piazza and discord; please use your preferred (recognizable) name on these platforms. You will need a secret key from our course submission server.

SCS students can access one of the designated labs for your course. The lab schedule can be found at: https://carleton.ca/scs/tech-support/computer-laboratories/. All SCS computer lab and 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**.

Course Work & Evaluation

Important Dates & Deadlines

Drill Practice	2:00pm of the Friday the week after associated lecture
Assignment 1	2:00pm on **Thursday Jan 27**
Assignment 2	2:00pm on **Thursday Feb 10**

Midterm Test (brightspace)	during your section's scheduled time on **Wed Mar 2**
Assignment 3	2:00pm on **Thursday Mar 10**
Assignment 4	2:00pm on **Thursday Mar 24**
Assignment 5	2:00pm on **Thursday Apr 7**
Final Test (brightspace)	TBD

Grade Computation

Drills	10%
Programming Assignments	60%
Midterm Test	12%
Final	18%

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 notes, lecture and workshop recordings, as well as links to assignments, drills, and any other coursework. You'll want to bookmark the page.

Drills (10%)

If you're a professional soccer player, you practice both match play with scrimmages and skills with focused drills (such as passing or shooting drills).

If you're a professional pianist, you practice both performance with dry runs through your piece, and skills with focused drills (such as practicing certain bars or one hand at a time.)

If you aspire to be a professional computer scientist, you should practice both writing larger and larger programs, and also the individual skills you need to write such programs.

The programming assignments in this course are an attempt to get you the former sort of practice--to simulate the kinds of tasks you may be asked to complete as a software developer. But as with soccer matches and piano performance, the "direct" task can be made easier and more intuitive, with skills practice. That is what these drills attempt to provide.

After each lecture, I will post 3-6 lecture-related "drill questions". These are meant to take a few minutes each, and are meant to either hone your intuition for the material, or give you technical practice. 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 homework assignment. Also, they are similar to test questions---in fact, 40% of the midterm and test questions are drawn directly from the drills.

There are ~22 drills (~1 per lecture). The best 16 are worth 10% combined; the remaining ~6 are dropped. Drills (on brightspace) are due by 2:00pm on Friday of the week following the associated lecture. Multiple attempts within the week are allowed.

Programming Assignments (60%)

The best computer scientists are the ones that have had the most practice. The programming assignments in this course are meant to give you the opportunity to practice with the topics of this course in a way that is challenging yet also manageable. 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. 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 your programs on your own. You may **not** use web search/the internet to look for approaches to the assignments; if you need help, we have many ways to do so without violating academic integrity..

There are 5 programming assignments, equally weighted. Programming assignments are due at 2:00pm on Thursdays on the relevant assignment's server (details on each assignment). Submit early and often to avoid last-minute technical issues (your best overall score is kept)**.** Lates are accepted within 24 hours, after which no lates are accepted due to posted solutions. If you have an emergency, please see the Late Policy & Emergencies section.

Midterm & Final Exam (12%, 18%)

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

- Studying for the exams hopefully improves your recall so that some of the more fundamental information can be recalled quickly.
- They encourage you to review all the course material, not just what is needed for the programming assignments and drills.
- They are the only coursework that evaluates you individually, since the remaining coursework can be collaborated on.

Having said that, both the midterm and the final are worth around as much as an assignment.

The midterm is worth 12% and the final 18%. The midterm and final will be multiple-choice questions on brightspace (40% of questions drawn directly from the drills). The final is cumulative. More information about the tests will follow on piazza, closer to their dates.

Late Policy & Emergencies

Drills must be submitted by the deadline, no lates accepted.

Programming assignments may be handed in up to 24 hours late without question, after which no lates are accepted, no exceptions (due to posted solutions.)

In the case of an emergency, you can, without explanation, not complete up to ~7 drills, and 1 programming assignment. If you need more accommodations, please contact Prof Alexa but be aware that dropping more marks is likely not possible. You need to do a minimum amount of programming practice in this course in order to deeply engage with the material. For this reason, **do not "squander" the emergencies for non-emergencies.**

Bonus (a.k.a. Extra Credit)

There are many opportunities for bonus points. If you don't use your lowest problem set grade for an emergency, or any of your lowest 7 drills, they are available for bonus. I recognize that if you have an emergency this might not seem fair, but the alternative is to offer emergency drops only to those that contact me (and that seems less fair and more cumbersome than offering the emergency drop to anyone who needs it without explanation.)

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. gain a deeper understanding of how data organization choices impact program performance, including that there is not one perfect data structure;
2. become comfortable reading, writing, and understanding algorithms, including analysis of their time and space efficiency; and
3. master the art of abstraction in order to reduce code complexity and increase code manageability and readability.

Necessary Equipment, Accounts & Software

You will need an internet-connected device to access lectures and notes, to access brightspace for drills and tests, to submit your assignments, and to access piazza and discord for Q&A and student hours.

You will need (free) accounts on piazza and discord; please use your preferred (recognizable) name on all of these platforms. You will be authenticated on both.

Finally, you will need a java compiler and your favourite editor (preferably IntelliJ as that is what our course staff is most familiar with and can therefore help you with).

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 problem solving tips sheet (posted on schedule)
- 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
- Attend or watch the assignment-specific workshops
- Go to Student 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.)
    - note that discord questions are the lowest-priority for course staff, as discord questions are often duplicates, and it's hard to know if they've been answered, etc. Please use piazza for questions you actually want answered.

Finally, note that there is no official support on the weekends, so please plan your time accordingly.

Academic Integrity

Assignments in this course involve coding, which benefits from bouncing ideas off of other people.

You **may** talk with peers and TAs at a high-level. But **you must write your code on your own**. If your solutions are basically line-by-line 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 programming assignments or solutions on the internet, before or after the due date.

Any students caught submitting copied code or overly sharing details of their code will 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 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

There is a **zero-tolerance policy in this course for disrespect of any kind**.

We expect you to treat your peers and course staff with kindness and grace. 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, and **you will be removed from course forums with no warnings**. If you have a problem with the course or course staff, the course forums are not a place to complain and whine. Prof Alexa will provide ~5 opportunities for you to give course feedback, so do not worry that your concerns will go unheeded. There is a time and a place for feedback; use it wisely.

You are required to read over our piazza posting etiquette as well as our discord #rules-please-read channel before using these forums. 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 there is anything Prof Alexa can do to help you succeed, please let her know as soon as possible so that she can accommodate accordingly.

Copyright

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 Prof 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.