Carleton University School of Computer Science

Introduction to Systems Programming

COMP 2401 A (Summer 2025)

Carleton University acknowledges the location of its campus on the traditional, unceded territories of the Algonquin nation.

1. Info at a Glance

Teaching Team:

- **Instructor:** Connor Hillen (Lecturer, He/Him)
  - **Email:** connorhillen@cunet.carleton.ca
  - **Office:** 5370 Herzberg
  - **Primary Communication:** Refer to the communication policy in Section 7 before reaching out.
- **Teaching Assistants:** A list of teaching assistants will be made available on the course website.
- **Student Hours:** A mix of online and in-person student hours will be scheduled and made available on the course website for both the instructor and teaching assistants.

Course Information

- **Lectures:**
  - **Timing:** Mon. Wed. 17:05 - 18:25
  - **Modality:** Hybrid, Synchronous
  - **Attendance** is expected to keep up to date with course information
  - **Recordings** will be posted, but several days later
  - **In-Person**: Refer to your schedule
  - **Online**: Zoom link posted to Brightspace page
- **Tutorials:**
  - **Timing:** Wed. 18:35 - 19:55
  - **In-Person:** Refer to your schedule
  - **Online:** Very limited online options are available for times you cannot make it to the tutorial; Zoom link posted to Brightspace page
  - **Weeks:** Tutorials begin the week of May 14th and are cancelled Weeks 7, 8, 9, 14, and 15.
- **Course Website:** https://brightspace.carleton.ca

Important dates and deadlines can be found here: https://carleton.ca/registrar/registration/dates/academic-dates/, including class suspension for summer break and statutory holidays.

1.1. Course Calendar Description

Introduction to system-level programming with fundamental OS concepts, procedures, primitive data types, user-defined types. Topics may include process management, memory management, process coordination and synchronization, inter-process communication, file systems, networking, pointers, heap and stack memory management, and system/library calls.

Precludes additional credit for SYSC 2006.
Prerequisite(s): (COMP 1006 or COMP 1406 or SYSC 2004) with a minimum grade of C-.

1.2. Course Outline Quick Links

2. Getting Started

Welcome to COMP2401! I hope that this course outline will be a helpful and easy to follow reference for the entire course. At any point that you have questions, always start by reviewing the course outline, and keep it on hand to easily reference throughout the course.

2.1. Course Materials

Note: Students are not required to purchase textbooks or other learning materials for this course.

2.1.1. Course Notes

COMP2401 Course Notes (2021 Edition), contributed to by Dr. Lanthier, Dr. Laurendeau, and Dr. Nussbaum. Freely available on the course Brightspace page alongside code examples.

What is it? These notes are a great resource, and we follow this edition of the notes pretty closely. They contain everything needed to achieve the learning outcomes of the course and additional code examples and supplementary information to help provide additional context to the material.

How to best use these notes? There will be some required readings throughout the term to prepare before following lectures, but otherwise it is recommended to review the relevant notes after lectures to fill in any knowledge gaps. You should **test out** the provided exercises and make sure you can follow along with the examples.

2.1.2. Course Virtual Machine

This course uses a **virtual machine** to provide a consistent work and testing environment for assignment and tutorial work. It is possibly the first course you will use virtual machines, but certainly not the last in the computer science curriculum.

We have a specific **virtual machine image** that has all of the software setup and configured to run for our specific version of the course. You will need to run this virtual machine to do your assignments and work on tutorials.

Why do I need to use a virtual machine? We need to make sure that specific versions of the software are being used because we are going to be working at a very low level, and small changes from one system to the next can make a big impact on what tools are available and what can lead to errors. We want all students working on a consistent environment so that your work works on the TAs computers the same way that it works on yours.

How much does it cost? Both VirtualBox and OpenStack are freely available.

2.2. First Steps

1. Review the full course outline and course plagiarism policies
2. Add the important dates of assessments to your personal calendar
3. Try to get Virtual Box running with the Summer 2025 virtual machine image.
   - **CLOSELY** follow the step-by-step guide, which includes both text and video guides, available here.
   - If the virtual machine image for the current term is not yet available, you can still test using an older version and start fresh with the new image once it is made available.
   - If you are experiencing difficulties, review the Virtual Machine Technical Support page which includes information about many common issues as well as instructions on how to receive in-person support.
   - If you are unable to run this, you can try using OpenStack - follow the guide here. OpenStack is a freely offered service hosted by the School of Computer Science where the virtual machine will run remotely on SCS servers. This will require an internet connection and use of a VPN when not on Carleton's campus.
4. Read Chapter 1 of the course notes and try to get a C program compiling and executing on the course virtual machine. Remember, any time you are reading the notes, stop and actually execute some of what you learn!
5. Familiarize yourself with the Brightspace page and layout
6. Say hello on the course forums! Check out Section 5.4 for information about how a forum post can earn some marks, but more importantly, help you meet your classmates!

3. Course Goals, Topics, and Outcomes

COMP2401 is a very important course in the CS curriculum. The goal of the course is to introduce you to the many system-level topics which underlie the programs that we write and execute.

This course is **not** a course about learning C. It is true that a lot of your time will be spent learning and understanding the C programming language, but it is important to understand this: The goal of this course is to **use C as a tool** for **exploring and understanding** the underlying systems concepts.

A full progression of the course is available in weekly calendar form with approximate topics, but the course tries to answer these questions as it progresses:

1. Representations of Computing and Memory
    1. What kinds of instructions does a processor execute?
    2. How is data represented in a way the computer can understand?
    3. How does a program get translated into instructions that can be executed by the computer?
    4. How is a program's memory organized and managed when running?
2. Interactions with the System
    1. How can our program request new memory while executing?
    2. What does it mean for a program to execute when there are multiple other programs being run?
    3. How can we write programs with multiple control flows?
    4. How can we write programs that can communicate with other programs?
3. How the Language Helps Use
    1. How can we use tools to improve the way our program translates into machine-code instructions?
    2. How does C know which data we are referring to when we use its name?

Throughout the course, it is **vital** to make sure that you understand the fundamentals before progressing. The essential goals of the course are not simply that you understand how to write a program in C, but that you can really understand how the C code translates into machine code, how the memory is organized, and generally build up an intuition about what programs are doing when they are executed.

We will **not** be covering assembly or hardware in-depth, and will be working at a slightly higher level of abstraction to begin.

If a student attends every lecture and completes every assignment and tutorial, then by the end of this course that student should be able to:

- Implement basic coding practices in the C language (e.g., loops, conditions),
- Selecting appropriate data types to solve problems in C,
- Design and implement static and dynamic data structures in C, such as linked lists,
- Execute builds of C programs using command line and Makefiles,
- Organize program structure to follow clean coding practices in C,
- Perform file input/output tasks,
- Implement concurrent programming techniques, including process management and threads

A full calendar of week-by-week topics will be made available on the course website.

4. Expectations and Succeeding in the Course

This may be the first time that you are thinking about computing and programs from a low-level perspective, which can be very tricky when you are first getting started.

It is important that you really take the time to understand the material before progressing. The topics all build on each other, even when it is not immediately obvious, and these topics are fundamental to future understanding of programming beyond this course. To succeed, we expect the following from students:

- **Master the fundamentals:** Make a point to understand the early material as best as you can to set a strong foundation. The earlier the content, the more it will provide vital understanding for later material. If you are struggling with the early materials, make sure to ask questions in class, ask questions on the official forums, do all of the exercises, read the course notes, and attend office hours.
- **Take responsibility:** Understand the course policies, understand that they will not be changed, and hold yourself to them. Going in with the expectation that policies are flexible may alleviate the feeling of responsibility for your own learning.
  - **What does responsibility look like?** Being proactive by starting assessments early. Reviewing materials after class. Double-checking your submissions. Learning and documenting deadlines. Reaching out for help when you don't understand something, and reaching out in the appropriate way. Accepting and owning mistakes or failures as they come up and turning them into opportunities for learning and to receive feedback. Taking advantage of Carleton's academic and personal support resources.
- **Experiment:** Reading the course notes or simply listening along in class are not enough; dig into the material. Identify connections between topics. Try to understand **why** we learned a topic. When reading notes, or during a lecture, or following the lecture, experiment with the provided examples. Do not simply read and assume you understand; come up with practice problems, set a goal, evaluate your understanding.
- **Engage:** Show up to class. Respond to polls and try to understand where you went wrong. Attend tutorials and use them as a chance for discussion instead of just getting a check mark. Complete assignments with the goal of learning what you do not know and growing.
- **Be kind:** We're all humans. A partner makes mistakes. A TA put the wrong mark after being up all night working on their own assignments. Your instructor wrote unclear or completely broken instructions. We **all** make mistakes and should be understanding and kind to each other. We will not tolerate abusive or hostile communication in this course.
- **Refer to the outline:** This outline has been filled with helpful information for you to refer to throughout the term.

4.1. Expectations from the Teaching Team

From us, we will be doing our best to be available for support, to fairly enforce our policies, and to provide helpful and honest feedback. We will attempt to provide plenty of practice and feedback for all materials.

Please feel comfortable reaching out to discuss suspected issues in provided materials. Please let us know if you require more feedback. If you feel there is insufficient practice available for content, let us know. We want to create a great learning environment, and we know that some things can improve.

Please also expect that we will not be bending rules unless we've otherwise violated our expectations. We will not change grading criteria to fit individual strengths, or provide accommodations beyond our course policies. Expect us to hold to our policies and to act as though you are taking responsibility for your own learning.

5. Assessments and Grading Scheme

5.1. Understanding the Course Assessment Structure

This course is designed to split activities into **formative** assessments and **summative** assessments. Formative assessments are here to **help you assess your own understanding**. They give you important practice with the materials. These frequently come in the form of practice exercises. Summative assessments are for **the teaching team to assess your understanding**. These are considered a **final assessment** of the material.

My philosophy: When possible, you should not be penalized for not knowing material during the first assessment. You should ideally be able to receive feedback and be given some opportunity to review that feedback, learn, and re-evaluate.

This means that in this course, only three assessments are highly weighted: Project 1, Project 2, and the Final Exam. All other assessments can be made up for by successfully completing those three assessments.

Does this mean I do not have to do them? I remember a math class in highschool where our teacher assigned us a bunch of exercises before our test. He called them “Bonus Exercises” for the test. A student asked, “So do we get bonus marks on the test for it?” and he simply responded, “Completing the exercises will lead to a higher mark on the test.” When asked again, “But how many exercises give us how many bonus marks?” he repeated himself. And you know what? Generally, those who did the practice exercises received higher marks on the test.

It is your responsibility to determine what is needed for your learning. In my experience, the more practice you get, the better your grade will be. Try not to treat the low-weight assessments as low-value; they are a chance to really try and experiment without stressing about the mark.

5.2. Assessment Scheme

Below is a table of the assessments and their default weighting, as well as approximate topics for each which are subject to change as the course progresses. Re-weighting and accommodations are described later.

ASSESSMENT	WEIGHT	DATE	TOPICS
Assignment 1	1%	Tue. May. 20	Chapter 1: C arrays, printing, basic I/O, functions
Assignment 2	1%	Tue. May. 27	Chapter 2: Binary, bitwise operations, data types
Assignment 3	1%	Tue. Jun. 3	Chapter 2, 3: Structs, strings, pointers
Assignment 4	1%	Tue. Jun. 10	Chapter 3: Dynamic memory
Assignment 5	1%	Tue. Jul. 1	Chapter 3, 4: Dynamic memory, linked lists
Assignment 6	1%	Tue. Jul. 22	Chapter 4, 5: Multithreading, Makefiles
Project 1	15%	Tue. Jul. 8	Assignments 1 - 3
Project 2	15%	Tue. Aug. 5	Assignments 4 - 6
Midterm Quiz 1 (Online)	7%	Fri. Jun. 20 - Sat. Jun. 21	Chapters 1.1 - 3.3
Midterm Quiz 2 (Online)	7%	Wed. Aug. 6 - Thu. Aug. 7	Chapters 3.4 - 5.4
Final Exam	40%	Scheduled by Registrar	Chapters 1 - 7
Side Quest Engagement	10%	Various	Practice

5.3. About Each Assessment

5.3.1. Tutorials

Tutorials are part of the "Side Quest Engagement" marks described in Section 5.4. They are an opportunity to do some helpful readings and exercises to get practice with the learning materials. You will also have an opportunity to discuss completed assignments with the teaching assistant and collaborate with classmates.

- Tutorials are intended to be completed in-person. Very limited online support may be provided, but should only be used when completely unable to attend.
- To get credit for tutorials, you will need to submit a code provided by the teaching assistant for engaging in the material for the majority of the tutorial session. You will submit to a quiz on Brightspace at the end of the session. If you are unable to attend in-person, you may also show a completed tutorial at the following tutorial (excluding the final tutorial) to receive the code for the previous week.

5.3.2. Assignments & Projects

Assignments are opportunities to practice the course material. They closely resemble the kinds of questions you will see on the projects. The marking scheme will be coarse and largely graded automatically. They are intended to provide feedback from the TA. Even if you miss a deadline, you should try to finish the assignment and check in during student hours to get feedback.

- **Submission:** Any submissions submitted after the penalty cut-off may not be reviewed depending on when the TA evaluates. Only submissions on Brightspace will be assessed. Submissions will receive a **5% deduction per hour** after the official deadline, up to a 100% deduction. You can still submit your assignment for feedback, but after the 100% deduction point, submissions may or may not be seen by TAs. The **latest submission** will be assessed.
- **Disputes:** If you wish to dispute marks for an assignment, you have **one week** from the announcement made on Brightspace about grades releasing to reach out using the Grade Disputes forum on Brightspace. Any disputes after this time will not be considered. Only disputes regarding marks incorrectly assigned based on the marking scheme will be considered.
- **Verify Your Submissions!:** It is your responsibility to download and test your submissions for the projects and assignments after you submit. Technical issues near the deadline are **not** grounds for extension or accommodation. Start early and submit periodically. Additional information about these kinds of considerations are noted in Section 8.
- **Projects:** These are larger, multi-week projects that can be optionally worked on in pairs. No use of AI is permitted as we must assess your understanding of the material directly. They will closely resemble the content covered in the assignments.
- **Automatic Re-Weighting:** Assignments 1-3 grades will automatically be replaced by Project 1 if you receive a higher grade on Project 1, and likewise for Assignments 4-6 and Project 2. **Note:** This is explicitly because the assignments have the same learning outcomes as the respective projects and are intended to be practice for that assessment.

5.3.3. Quizzes

To remain flexible, the quizzes are **asynchronous, online**. Class is **cancelled** during quiz dates so that you may optionally write the quiz during class time. The quizzes will be unproctored and held on Brightspace. They are an opportunity for you to evaluate your own understanding before the final exam.

Automatic Re-Weighting: Any lower quiz marks will be replaced by the final exam. This is explicitly because the quizzes are intended to act as practice for the final summative exam and cover the same outcomes.

5.3.4. Final Exam

The final exam is held **in-person** and is scheduled by the registrar's office. It will cover all of the material in the course with special attention to the materials covered by Quiz 1, Quiz 2, and the materials following Quiz 2. It is multiple choice and closed-book.

5.4. Side Quest Engagement

Side Quests are an opportunity to make up some marks by engaging in course material. Side Quests are effectively a way to gain extra credit to supplement project and final exam scores by engaging in activities designed to provide extra practice or to engage in helpful learning tasks that are closely associated with the course and curriculum goals. There are more quests available than marks, and all unfulfilled quest marks will be re-weighted to the summative assessments.

Re-Weighting: If you do not complete all or any quests, you will **not** be penalized; instead, this will only increase the weight of the summative assessments (Projects, Final Exam) proportional to the missed marks. E.g., if you complete 2% of your total 10% side quest marks, the remaining 8% will be distributed between Project 1, Project 2, and the Final Exam. Because the final exam is worth more than each project, the final exam will increase in weight more than each project.

How do I submit side quests?

- Any side quests requiring a submission must be submitted to the "Private Side Quest Submissions" forum on Brightspace.
- Some side quests do not require a forum post submission as they will be assessed elsewhere
- Side quests are graded SAT/UNSAT and you may be given a chance to re-submit
- **Note:** Brightspace will not be able to display quest grades very consistently.

What if I have a really cool idea for a quest?

- Let me know! If the quest is relevant to the course goals, I may make it available for everyone to submit. Otherwise, only the official quest marks are available.

What side quests are available?

An updated list may be posted to the course website, and weights may be slightly adjusted with an announcement made, but the following quests are available right away:

- **Tutorials (0.5% x 9):** Each tutorial is worth **0.5%** toward your side quest marks. Tutorials are **highly suggested** as they provide essential readings and exercises for understanding the course materials. Assessed by submitting a code to Brightspace provided by the TA if you have engaged throughout the tutorial. You must attend and engage for most of the tutorial to receive your mark.
- **Say Hello (0.5%):** There is a "Say Hello!" forum on Brightspace with instructions needed to follow where you can say hello to me and other students to get to know each other.
- **In-Class Polls (2% x 2):** We will have Wooclap polls in each lecture where we will take attendance and have questions to review and learn about material. There are two opportunities to receive marks: Before the break and after the break. Completion of approximately 80% of the polls before the break will earn 2%, and approximately 80% of the polls after the break will earn an additional 2%.
- **Weekly Quest Quizzes (1% x 3):** There are weekly quizzes on Brightspace which are designed to test your understanding of the course material from the week before and possibly test some understanding of upcoming material based on readings. You receive 1% for every **three quizzes** which receive a **>50% mark** after **three attempts**. To receive the full 3%, you must complete 9 quizzes.
- **Assignments and Projects:** Assignments will each have bonuses available, worth 0.5% each unless otherwise specified. Projects will have at least 2% worth of quests associated with them, specified on the project.
  - **Git:** At minimum, each project has a 1% quest for using Git as version control during the project. You may wish to get practice with Git on assignments before the projects if you would like to receive this quest.
- **More?** Additional quests may be posted to Brightspace over time.

Not sure the best route for you? consider that attending all tutorials (4.5%), saying hello on the course forums (0.5%), attending class and responding to in-class polls (4%) and completing just three weekly quizzes (1%) is enough to satisfy all of the quest marks. Completing the weekly quizzes each week can provide some buffer, just in case you miss a few lectures or tutorials. This doesn't mean you can't still challenge yourself with assignment and project quests, though!

6. Missed Work Policies

6.1. Assignments, Quizzes, Side Quests

No further accommodations beyond the existing re-weighting scheme will be applied, but feedback can be provided in student hours if you cannot submit in time for a TA to evaluate and provide feedback. Quizzes will be re-opened so that you can complete it and receive feedback, but they will not be used for marks.

6.2. Projects

First, review the official Academic Consideration Policy (https://carleton.ca/registrar/academic-consideration-policy/) to make sure that you fall under appropriate conditions for accommodations. Reach out to the instructor noting how many days you were incapacitated and prepare to fill out the short or long-term accommodation form as needed, but do not fill it out until contacted by the instructor. Reach out as soon as you are can, at most **24 hours** following the due date of the project (unless circumstances prevented reaching out in time).

Short-Term (5 days or less): A short-term extension may be available. Requests near the deadline may be accommodated on short-notice; be prepared to submit on short notice once an extension is provided. You may also email a copy of your project if accommodations are not provided early enough and it may be considered based on the provided timeline.

Long-Term (>5 days): Typically, long-term accommodations will require submission of the material before the end of the term. The submission will receive a SAT/UNSAT mark from the instructor and a SAT submission will have the weight re-distributed proportionally to the other formative elements. Only **one** project can have long-term accommodations applied.

6.3. Support

Feeling Sick? If you are feeling very sick (e.g., fever, chills, stomach upset) please do not come to campus. If you have missed lectures, recordings will be available. If you have missed tutorials, the materials will be available and you can either attend another section after reaching out to the instructor (policy may change, check Carleton emails for information) or attend TA or Instructor office hours to review the missed material.

Mental Health Concerns? If you are struggling, please do not hesitate to reach out to me. I am happy to listen, or even to just provide/direct you to resources that might help. If class work is overwhelming, check out the support resources below and consider attending office hours with the instructor or TAs to try and catch up. Carleton offers a wide array of mental health resources, and I encourage you to take time to review them: https://carleton.ca/wellness/

Help with Course Materials: You can expect to spend about 8 hours per week on this course, in addition to lecture time. If you find yourself spending a very long time with assignments, feeling like you've missed important parts of the course materials and are getting lost, or otherwise are struggling with the material, support is available!

- **Requirements are unclear!** Check the forums to see if clarifications have been made and make a post if nobody has asked this question. Asking publicly helps clarify assignment requirements for everyone!
- **I don't understand a concept!** Review the course notes and recommended readings and make sure you have tried the available exercises. Bring specific examples to a TA in office hours if you can attend, and if you can not clear it up during office hours, go to instructor office hours to discuss. If the times do not work, email the instructor to try and set up a meeting. Tutorials are also a great time to ask peers how they understand the concepts, as they may have overcome the same problems you are encountering or you can tackle it together.
- **I'm concerned about workload!** Reach out to the instructor via email and I would recommend emailing our undergraduate advisors to discuss more.
- **General academic skills support?**
  - Science Student Success Centre: https://sssc.carleton.ca/
  - Carleton Computer Science Society Events: https://ccss.carleton.ca/events/
  - Centre for Student Academic Support: https://carleton.ca/csas/

7. Communication Policy

Communication that does not meet this policy may not receive a response, or may have their response delayed until an announcement is made.

Students are expected to check their Carleton email addresses daily for announcements. Reminders for upcoming assignment deadlines will be handled using the default Brightspace notifications system, so if you require reminders, check that your Brightspace settings will send notifications.

Most communication should be handled in-class, on the student forums, or during student hours. If you miss information from a lecture, review available recordings or reach out to another student in-class or on the forums.

- If materials, policies, or assignment requirements are unclear, post to the forums or attend student hours.
  - **Note:** Student hours are not intended to be used to help debug code, but to better understand materials, receive academic advice, and to receive feedback on exercises
- If you are having **technical issues** with the virtual machine:
  - Carefully following the setup guide step-by-step for Virtual Box available here
  - If you cannot run Virtual Box, get setup with OpenStack by following the guide available here and/or use the School of Computer Science computer labs
  - If you are experiencing technical issues, review the technical support guides available here
  - If you are unable to find a solution, post the the Brightspace forums and attend TA student hours noting the steps that you have followed
  - If neither of these resolves your issues, consult the technical support procedures available here
- If you require accommodations or some other personal request that requires personal information:
  - Email the instructor
  - Include your name and student ID in the email
  - Include the course code in the subject line
  - Include all necessary information
- If you are unsure of dates or deadlines, consult the places where dates and deadlines are posted (calendars, specification, course outline) and only reach out on the forums if something is misaligned
- Expect up to **3 business days** to receive a response before following-up. This means you should plan ahead and make sure that if you have questions or concerns you can ask them early.
- If you would like to dispute a grade:
  - Post to the private grade dispute forum on Brightspace within one week of the grade release
  - Include all information about the issue with the mark
  - **Note:** Only instances where an incorrect grade were provided will be considered.
- **Emails** about information that is publicly available in the outline or Brightspace page, or that should have been asked in forums, or has been recently clarified in lectures, may be responded to **in-lecture** or **in an upcoming announcement** rather than an email response so that all students receive clarifications. Make sure to check-in on lectures and announcements to see if your question has been responded to there.
- **Students MUST behave in a professional manner in all communications.** Any communication that is seen as abusive or harassing will be moderated, ignored, or reported to the university for disciplinary action.

8. Important Considerations

1. **Technical Issues are NOT grounds to overrule policies.** Make sure to review these considerations. The computer labs are available to work in if you experience technical issues with your computer and you should give yourself enough time to utilize these before the deadline if issues arise.
2. **Test and Verify Submissions:** It is your responsibility to download and test your submissions after submitting to make sure that they work as intended and that all files were correctly uploaded.
3. **Backup and Submit Work Often:** In COMP2401, it is especially easy to accidentally delete or corrupt your files. One way to maintain backups of your work is to submit often to Brightspace, utilizing version control tools like Git and online repositories like GitHub, or utilizing the Carleton and SCS provided Microsoft OneDrive or NextCloud services. This ensures that if you experience technical issues, you can still recover your work to continue. Learn more here: https://carleton.ca/scs/tech-support/backups/
4. **Assignments Need Functioning Code:** Code which does not compile or execute can be subject to heavy penalties, up to and including a **zero**. It is an expectation that you will submit something functional at minimum. It is often better to submit something partially complete that compiles and executes properly than something that seems more functionally complete, but can not be run to verify this.
5. **All materials created for this course remain the intellectual property of the instructor:** These materials are intended for the personal and non-transferable use of students registered in the current offering of the course. Reposting, reproducing, or redistributing any course materials, in part or in whole, without the written consent of the instructor is **strictly prohibited**.

9. Course Scheduling and Modality

COMP2401 A will be held using in-person lectures that **MAY** be recorded, but recordings are **NOT** a guarantee. A Zoom link will be made available for hybrid attendance if you are unable to attend in-person (e.g., due to illness). Lectures may be adjusted to be held online at the Zoom link if the instructor is ill or in cases of severe weather; an announcement will be made ahead of the lecture under these circumstances.

Synchronous attendance is expected. Importantly, many students say that they will rely on recordings instead of attending, but end up not watching the recordings, falling behind, and not being able to ask important clarifications during lectures. Try to use recordings only to review or to make up for missed lectures. If you do not attend a lecture, it is your responsibility to reach out to fellow students to receive missed information for that lecture before attending the next lecture if the recording is not immediately available.

Please note that by participating in these lectures that you may be included in these recordings. When attending on Zoom, Zoom will always notify meeting participants that a meeting is being recorded. It is not possible to disable this notification. These recordings will only be available to the members of this class, and I ask that everyone be respectful and not allow others to view the recordings. At the end of the course, the recordings will be deleted.

Please note that recordings are protected by copyright. The recordings are for your educational use, and you are not permitted to publish to third party sites. If you have concerns about being recorded, please email the instructor directly so we can discuss these.

10. Plagiarism Policy

10.1. Why Worry about Misconduct?

Academic integrity is important. Typically, your work is analyzed by standardized tools to detect likely cases of collaboration, posting materials online, or using solutions from online. If you are not caught, it may seem positive: Receiving a higher grade than you deserved. You may feel like you have a good justification, and that it does not harm anyone else; however, misconduct can hide other issues that should be addressed. If my material is too challenging or too overwhelming, I want to fix that. If my lectures are too confusing to learn, I want to fix that. If life events get in the way of your work, I want to make sure you do not fall behind.

If you do not demonstrate your own knowledge, it hides these issues from us. You may end up unprepared for future courses which rely on this knowledge or less likely to ask for the help you need. It is important to follow the policies because it is right, because it is a requirement, and because it can incur serious penalties when caught, but this is me – Connor Hillen – asking you to consider the broader reason we ask you to complete these materials in the way we ask. We want you to learn, to become a professional and leader. That means taking responsibility for both successes and failures, even when it is challenging. If you are struggling with material, or time, or academic skills, or don't understand the purpose of the materials, please let me know.

10.2. Policies

If you are unsure of the expectations regarding academic integrity (how to use and cite references, if collaboration with lab- or classmates is permitted (and, if so, to what degree), then you must ask your instructor. **Sharing assignment, project, or quiz specifications** or posting them online (to sites like Chegg, CourseHero, OneClass, etc.) is always considered academic misconduct (at any time, **even after the course has concluded**).

You are **never permitted to post, share, or upload course materials** (even for portfolio purposes, e.g., a public GitHub repository, Stack Overflow) without receiving explicit permission from your instructor. Academic integrity offences are reported to the office of the Dean of Science. Information, process and penalties for such offences can be found on the ODS webpage: https://science.carleton.ca/students/academic-integrity/.

Additionally:

1. All work must be completed individually unless explicitly stated on the specification,
2. Where collaboration is permitted, all collaborator names and contributions must be cited in comments (e.g., over functions, top of a file) and in supporting materials (e.g., README files),
3. Never pass off work from any other source as your own - if you received detailed assistance from a permitted source, cite this source in the comments (e.g., course notes pages, lecture, TA),
4. You may only use the resources explicitly stated in test descriptions during quizzes, tests,
5. You are **never** permitted to help someone commit plagiarism: distributing your rough or final work, work others have written, or making it easy to acquire your own work (e.g., leaving an unlocked laptop with other students),
6. Use of Generative AI (e.g., ChatGPT, Copilot, Llama) is **permitted** for Assignments 1-6 and tutorial exercises, but the transcript of conversations or description of generated code must be included, any areas of code must be cited using the rules above, and you must be prepared to explain the code if it is unclear if you understood the generated results.
7. Use of Generative AI is **NOT** permitted during the projects, tests, quizzes, or midterms, or any other assessments unless explicitly stated. Please read Section 11 for information about *why* these policies are in place.

11. Generative AI

Conversational and Code Generative AI is a very new and evolving area in programming, computer science, and education. Personally, I believe it can be very helpful for learning and supporting our work, but I have also seen first-hand that improper use of generative AI can unintentionally hinder learning. There are many ethical problems to consider, which have led some to question if generative AI really is the future of work in our field:

- The high power consumption of training and operating large language models,
- The collection of private and confidential data by large companies,
- The unlicensed use of other people's intellectual property to create an AI product to sell,
- A possible over-reliance of generated code which might be used in safety critical systems without proper vetting,
- The challenge of differentiating made up "hallucinations" from real information,
- Unintentionally using generative AI to reach solutions without understanding what got there, and thus, not being able to generalize that knowledge for future problems and learning,
- Equitable access to generative AI, which can be costly,
- The possible reduction of skilled, talented, and knowledgable professionals from workplaces, or otherwise worse working conditions and bargaining capability.

As such, some companies ban the use of generative AI, or refuse to sell products made with generative AI.

From an educational perspective, there are some ways to work with the AI that can be beneficial, and many ways it can be harmful. It is important to recognize that learning requires some amount of challenge: Things will be difficult and not work out right away, and then through trying, gaining information, getting feedback, and trying again, you learn.

Generative AI is permitted under some circumstances (see Section 10.2), but even then, please keep in mind:

1. A future employer / class may not permit use of AI, so make sure you really know what you are doing!
2. Finding any way to get the right answer is **not** the goal. We want you to understand the content; the answer helps us now if you are on the right track. I, as your instructor, am not trying to learn if Llama 3 knows how to code a function. I want to make sure you can and are prepared for your next courses.
3. Using AI for repetitive, mindless tasks is great, but can still reduce your understanding of fundamentals!
   - Generative sample data to test your program,
   - Making a minor change across lots of different areas of code,
   - Repeating the same line over and over again but with different parameters sent in.
4. Getting analogies from AI to help understand a concept *can* be great, or terrible:
   - Asking for an analogy, or different levels of explanation, can be very helpful,
   - … however, you might receive an entirely incorrect analogy that **sounds** good it is challenging to unlearn.
5. Getting AI to explain a section of code can be very helpful!
   - But it can also explain it incorrectly in a very convincing way

All-in-all, I suggest using AI sparingly for your work where it has been permitted by our plagiarism policy. Receiving explanations, using the AI to explain confusing concepts from class, using AI to create exercises for you to gain practice, these can all be really helpful - but you should also use the knowledgeable teaching team to make sure you have accurate information, always make sure that work that you submit demonstrates **your** understanding and not an AI's output, and avoid becoming reliant on specific tools to build your foundational knowledge in your profession.

12. School of Computer Science Information

School of Computer Science Laptop Requirement: Every student that has been enrolled in a 1000-level (i.e., first year) course offered by the School of Computer Science after the 2020/2021 school year is required to have a laptop. This includes COMP1001, COMP1005, and COMP1006. For more information, please visit https://carleton.ca/scs/scs-laptop-requirement/ and then review the requirements at https://carleton.ca/scs/scs-laptop-requirement/laptop-specs/.

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

SCS Computer Laboratory: Students taking a COMP course can access the SCS computer labs. The lab schedule and location 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/tech-support/. Technical support staff may be contacted in-person or virtually, see this page for details: https://carleton.ca/scs/tech-support/contact-it-support/.

13. Academic Accommodations and Regulations

Academic Accommodation: Carleton is committed to providing academic accessibility for all individuals. You may need special arrangements to meet your academic obligations during the term. The accommodation request processes are outlined on the Academic Accommodations website (https://students.carleton.ca/course-outline/).

Academic Integrity: Students are expected to uphold the values of academic Integrity, which include fairness, honesty, trust, and responsibility. Examples of actions that compromise these values include but are not limited to plagiarism, accessing unauthorized sites for assignments or tests, unauthorized collaboration on assignments or exams, and using artificial intelligence tools such as ChatGPT when your assessment instructions say it is not permitted.

Misconduct in scholarly activity will not be tolerated and will result in consequences as outlined in Carleton University's Academic Integrity Policy. A list of standard sanctions in the Faculty of Science can be found here.

Additional details about this process can be found on the Faculty of Science Academic Integrity website.

Students are expected to familiarize themselves with and abide by Carleton University's Academic Integrity Policy.

Student Rights & Responsibilities: Students are expected to act responsibly and engage respectfully with other students and members of the Carleton and the broader community. See the 7 Rights and Responsibilities Policy for details regarding the expectations of non-academic behaviour of students. Those who participate with another student in the commission of an infraction of this Policy will also be held liable for their actions.

Student Concerns: If you have any concerns regarding this course, your first point of contact is me. Please email me or visit during my student hours, and I will do my best to address your concerns. If I cannot resolve the issue, the next point of contact is the School of Computer Science at studentconcerns@scs.carleton.ca. If the concern remains unresolved, the final point of contact is the Office of the Dean of Science at ODScience@carleton.ca. Please follow this order of contact.

Note: You can also bring your concerns to Ombuds services.