

Land acknowledgement: We, the people of the Faculty of Science at Carleton University, acknowledge that our campus is located on the traditional, unceded territories of the Algonquin Anishinabeg people. Miigwetch for your hospitality and stewardship of this territory and the teachings that come from it. We are grateful for this land, the air that we breathe, and the water that sustains us all as well as for the animals, plants and other living beings: these enable us to research, teach, mentor, support, study, and learn. We recognize our responsibility to our natural environment and to reconciliation with Indigenous peoples.

Teaching Team:

- **Instructor:** Connor Hillen (Assistant Professor, He/Him)
 - **Email:** connorhillen@cunet.carleton.ca
 - **Office:** 5370 Herzberg
 - **Primary Communication:** Refer to the communication policy below before reaching out.
 - **Call Me:** Connor (pronounced 'kon -ur'), Professor, or Mr. Hillen (pronounced 'hill -len') if you prefer
- **Lab Coordinator:** Vojislav Radonjic He/Him
 - **Email:** vojislavradonjic@cunet.carleton.ca
 - Primary contact for information about TA concerns, labs, and marking.
- **Teaching Assistants:** A list of teaching assistants will be made available on the course website.
- **Student Hours:** Instructor student hours will be posted on the course website beginning Week 2.
- **Course Website:** <https://brightspace.carleton.ca>

- **Lecture Locations:** In-Person, refer to your schedule
- **Lecture Times:**
 - **A:** Tue., Thu. @ 08:35 - 09:55
 - **B:** Tue., Thu. @ 14:35 - 15:55
 - **C:** Tue., Thu. @ 10:05 - 11:25
- **Tutorials:** Beginning Week 2
 - **Location:** In-Person, refer to your schedule.
 - **Times:**
 - **A1:** Mon. @ 16:05 - 17:25
 - **A2:** Mon. @ 19:35 - 20:55
 - **A3:** Mon. @ 10:05 - 11:25
 - **B1:** Tue. @ 08:35 - 09:55
 - **B2:** Tue. @ 18:05 - 19:25
 - **B3:** Mon. @ 14:35 - 15:55
 - **C1:** Wed. @ 08:35 - 09:55
 - **C2:** Tue. @ 19:35 - 20:55
 - **C3:** Tue. @ 11:35 - 12:55

Important dates and deadlines for the academic year can be found here: <https://students.carleton.ca/academic-dates/>.

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 and is not equivalent, unless taken prior to Fall 2018. Also precludes additional credit for SYSC 1006. SYSC 2004 (with C- or above) may only be used as a prerequisite if taken prior to Fall 2025.
Prerequisite(s): (COMP 1006 or COMP 1406) with a minimum grade of C-.

Calendar prerequisites are strictly enforced; this includes students writing deferred final exams. For details, see the SCS Prerequisite Policy: [Prerequisites](#).

1.1. Course Outline Quick Links

It is your responsibility to read the full course outline carefully, but here are some quick links so that you can easily refer to them throughout the course if you have questions:

- [\[Section 5\]](#) Assessment Scheme and Grading
- [\[Section 5.5\]](#) Accommodations and Missed Work Policies
- [\[Appendix A\]](#) Engagement Task List
- [\[Section 5.6\]](#) Technical and Other Support Resources
- [\[Section 6\]](#) Communication Policy, Grade Disputes, & Receiving Help
- [\[Section 8\]](#) Important Considerations, Additional Policies, and Expectations
- [\[Section 9\]](#) Plagiarism and Academic Integrity Policy
- [\[Section 10\]](#) AI and Generative Tools Information

2. Note from Connor

Hello and welcome to COMP2401! No matter your background with the material, this course should feel like a place where you are supported and asking questions is normal. You are not supposed to have all the answers when you arrive. Many students secretly worry that they are “uniquely” behind; this is simply untrue. Many folks struggle with the material and this helps deepen understanding and the TAs and myself are available during student hours and lectures to help you get on track.

This course is meant to be a welcoming and accessible space for students of all backgrounds, identities, and experiences. I recognize the different paths bringing folks into the course as valid pathways that might need different supports. If any part of the course is creating a barrier for your learning or participation, please reach out to me to find a way forward. You are welcome to tell me your preferred name and pronouns that you would like me to use.

On Learning: When AI and solutions are just a text box away, *why bother struggling with a problem when the faster solution is right there?* While often it can be considered misconduct, personally, I find the worse result to be that you lose out on **understanding**. Understanding comes from productive struggle and is needed to build a foundation for bigger ideas.

We want you to be more than the middleware for AI and search engines. It may not be obvious right now, but every calculation, compilation error, and line of code written builds up muscle memory for what eventually becomes intuition and the “aha!” moments that separate you from someone who didn’t have this learning environment.

Succeeding in this course doesn’t mean a high grade, it means understanding the material enough to connect it to what you have already learned and what you will learn in the future. Be careful not to skip past the confusing parts; review old notes, chat with a classmate, revisit old exercises. Challenge yourself a bit, then if you haven’t made progress, reach out on the forums, in tutorials, or during student hours. Sometimes your brain just needs a break from the heavy cognitive load and a bit of rest will help connect those ideas. Start early, leave time for breaks, and remember that learning happens over time.

3. Learning Material and Other Course/Lab-Related Resources

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

- **Course Notes:** COMP2401 Course Notes (2021 Edition) by Dr. Mark Lanthier with contributions from Dr. Christine Laurendeau and Dr. Doron Nussbaum.
 - Available for free on the course Brightspace page alongside code examples.
- **Lecture Slides:** Starting Week 2, you will be expected to review some readings in advance. Lectures will not follow the slides one-to-one and will primarily consist of Woodclap Q&A, clarifications, and exercises. Slides in the Weekly Slides that are primarily expected to be readings prior to the lecture have “R” in front of their title (e.g., **R3.1 - Data Types.pdf**).
- **Virtual Machines:** This course requires the use of a virtual machine to ensure that you get an opportunity to learn about working with Linux and most importantly to ensure that your code runs consistently on both your computer and your TA’s computer. The virtual machine runs a **headless** distribution of Linux, meaning that there is no traditional user interface, and instead you are encouraged to learn about development using the terminal.
 - **Downloads & Information:** <https://carleton.ca/scs/tech-support/virtual-machines/>
 - **Image:** Download the appropriate VM image for this course and term when it is made available at the link above.
 - **Virtual Box (Free):** First try to use Virtual Box, allowing you to run the virtual machine on your own computer [\[Guide\]](#)
 - **OpenStack (Free):** Optionally (or if you run into issues with Virtual Box) you can try running the virtual machine on the School of Computer Science cloud service. Connecting will require the free Carleton VPN. [\[Guide\]](#)
 - **Labs:** The virtual machines are also installed on machines in the SCS labs.
 - **Expectations:** You are expected to learn to use the VM on your own. The instructor will not be providing technical support. You should take time to read the technical support pages linked above **carefully** to get operational with the VM. The first tutorials and TA student hours can help to get started if you encounter issues.
 - **Using the VM:** Because the virtual machine is headless, you will likely want some way of connecting to the machine using SSH (Secure Shell). You are encouraged to use a terminal to connect to this (e.g., PowerShell, PuTTY), but you can also use Visual Studio Code to connect to your workspace using the official *Remote Explorer* extension.

4. Topics Covered and Learning Outcomes

A full calendar for weekly course topics can be found on the course Brightspace page, along with required readings

A common misconception is that this is just a C programming course. C is used to explore lower-level ideas without going all the way down to the assembly or machine code level. The high-level goals for this course are to build an understanding of:

1. Computation and Data Representation in Memory
 - **Define and Discuss:** Data types, areas of program memory, binary and hexadecimal representations, memory addresses, pointers, virtual memory
 - **Calculate:** Conversions between hex, decimal, binary, octal; estimated memory usage of data structures; binary representations of different data types in memory
 - **Implement:** C programs using dynamic memory, custom data types, bitwise operations, programs with pointers, linked lists, and dynamically allocated arrays
2. General Execution of Individual Processes
 - **Define and Discuss:** Process management, control flows and threading, system calls, basics of process scheduling, communicating between processes using TCP, UDP, and signals, streams, buffers, the difficulties of concurrency
 - **Implement:** Multi-threaded programs, multi-process systems, binary and text-based file I/O
3. How the C Language Translates into Executable Code
 - **Define and Discuss:** Compilation from source code to object code, linking object files into executables, C libraries
 - **Implement:** Makefiles, multi-file projects

4.1. Student Expectations

In order to meet these outcomes and be prepared for your upcoming courses, research, or industry work, here are some of the expectations that we have for students:

1. Attend lectures and take notes. If you miss one, catch up on any missed information with a peer in the class.
2. Read the course notes and required readings. When you see code examples, take time to run and play with them.
3. Submit assignments on time. Always review the requirements before submitting. Give yourself time to download your submission and run it on the course virtual machine to verify that what you submitted works as expected.
4. Attend student hours if you feel like you're struggling, taking too long on assignments, or want to go deeper into a topic.
5. Before reaching out to ask about policies, dates, and deadlines, review the course outline and provided materials.
6. Only submit work that you can explain. Do not submit work that was completed completely at random, that was completed by an AI, or was completed by someone else. The goal is understanding, not just getting a functioning result.
7. Attend tutorials to collaborate with peers and have discussions about the material. If you miss a tutorial, find time to read it and try out the exercises. Consider coming to TA or instructor student hours to go over it if you find it confusing.
8. Read the feedback on your assignments and give the second-chance a try once you learn how to improve your work.
9. Communicate with the teaching team and with other students in a kind and respectful way.

4.2. Teaching Team Expectations

Here are some expectations that you should hold the teaching team accountable for:

1. We will attempt to reply to messages within about 2-3 business days, or else we should receive a follow-up.
2. We will reply to communications in a kind and courteous way.
3. We will aim to grade accurately and fairly in a way that aligns with our grading schemes for all students.
4. We will periodically request anonymous feedback to help us improve the teaching of the course.
5. We will show up on time and prepared to teach and support the students.
6. We will do our best to help you with your concerns during student hours, labs, and on the forums, but when an inquiry needs more information or time, we'll reach out to folks who can help us better understand the best approach to help.

5. Assessment Scheme

The topics in this table are subject to change, per the needs of the course.

ASSESSMENT	WEIGHT	DUE DATE	APPROXIMATE TOPICS
Assignment 0	Optional (See Section 5.4)	Sep. 24, 23:59	Ch. 1, 2.5: C basics, arrays, user input
Assignment 1	5%	Oct. 08, 23:59	2.1–2.2, 6.1–6.2: Bitwise operations, binary file I/O
Assignment 2	5%	Oct. 22, 23:59	2.3–2.6, 3.1: Pointers, strings, compound data types
Assignment 3	5%	Nov. 12, 23:59	3.3–3.5, 4: Dynamic memory, collections, Makefiles
Final Project	15%	Dec. 07, 23:59	Chapters 1–6: Summative (incl. concurrency)
Assignment 1 Term Test	4%	Oct. 01, In-Class	C basics, compilation, bitwise & binary I/O
Assignment 2 Term Test	4%	Oct. 15, In-Class	Pointers, strings, compound data types
Assignment 3 Term Test	4%	Nov. 05, In-Class	Dynamic memory, areas of memory
Project Term Test	4%	Nov. 26, In-Class	Concurrency, processes, IPC
Final Exam	40% (40% minimum score required to pass)	Scheduled by Registrar	Chapters 1–7
Engagement Tasks	14% (+2.5% bonus)	Various, latest submission: Dec. 11	Practice

Final Grade Rounding Policy: Final grades will not be rounded, as this applies an uneven boost to different students. Instead, all students receive 0.1% added to their final grade to account for any minor rounding errors during calculation.

Test Information: Term Tests and Final Exam are written in-person and are Multiple Choice. Each Term Test covers the course material that is relevant to the associated assignment and may include some additional topics from earlier in the course. Your feeling about the material after writing should help to inform whether you need some additional conceptual support via student hours or course note review to help get comfortable starting the assignment.

40% Exam Hurdle Requirement: It is important to take studying and feedback during the term seriously. In addition to the standard course grade requirements, you must receive a minimum grade of 40% on the final exam to pass the course. A student who scores below 40% on the final exam will receive a grade of F in the course, regardless of their computed grade.

Practice, Feedback, Revise, Test: This course does not expect you to get everything perfect on your first attempt. You learn through difficulty, through trial and error, through feedback, and through applying feedback. I do not want you to feel compelled to commit misconduct or otherwise hinder your learning in an attempt to get a perfect grade on the first try.

- 1. Assignments and Term Tests:** These are for *practice* and *information gathering*. You will have multiple attempts to submit these to improve your mark as well as replacing the marks with the later assessments, so completing tests and assignments are all about helping you gauge your understanding and let you know if you need more help.
 - **Always** try to submit something during the first attempt window. Statistically, those who tend to skip the first attempt and only submit to the second-attempt windows tend to also perform worse on the final exam. Even a partial submission will get you helpful feedback.
 - Give it your best shot. If you perform poorly, this is an opportunity to attend student hours to get back on track.
- 2. Project and Exam:** These are for *validating understanding*. Take these seriously and recognize that the other course elements are practice for these. These marks can replace the relevant assignments and tests per the reweighting policies ([Section 5.1](#)). These marks are final and gauge your final understanding of the material from the term.
 - There will be little leniency provided on these assessments. Use the other course materials to help you adequately prepare for the finals.

5.1. Re-weighting Policy

- A higher score on the Final Project will automatically replace any lower Assignment 1 - 3 scores,
- A higher score on the Final Exam will automatically replace any lower Term Test scores,
- **Note:** Not completing an assignment/term test in anticipation of this policy is NOT recommended as this effectively increases the weight of the finals and leaves your grade more vulnerable to issues encountered at the end of the term.

5.2. Second-Chance Policy

To promote learning through mistakes and feedback, this course will be permitting the re-submission of Assignments 1–3 and re-taking of the term tests. The final mark for the assignments and term tests will be based on the *Weighted Average with Insurance* scheme that is described below.

- **Assignments 1–3:** After you receive feedback for an assignment, a “Second Chance” submission page will open on Brightspace for **one week** where you can choose to submit an updated version of your assignment with revisions.
 - Grading times may have some variability, so the re-submission deadline is not posted in advance and is based on the date that grades are released,
 - The second submission will receive less feedback than the first submission to expedite grading time,
 - The overall mark for each assignment uses the “Weighted Average with Insurance” scheme described below.
 - Your resubmission **MUST** include a file called `CHANGELOG.txt` or `CHANGELOG.md` which, in bullet-point form, describes the changes that you made from your first submission, if you submitted one. This explanation cannot be AI-generated. Submissions that do **not** include a `CHANGELOG` may receive a zero.
- **Term Tests 1–4:** One week after the in-person test, an online quiz will open on Brightspace that permits you to optionally rewrite the test as a “second chance” submission as many times as you wish. The second-attempt mark will be the highest of your online attempts. These will also serve as the primary review practice for the final exam.
 - The overall mark for each quiz uses the “Weighted Average with Insurance” scheme described below.
- The final project and final exam are **not eligible** for re-submission or re-taking and are intended to be the final evaluation of your understanding of the course material.
- **These policies are also the short-term accommodation for students who miss the original deadlines.** Make-up marks are available via Engagement Tasks ([Section 5.4](#)).

Weighted Average with Insurance: The final, computed grade (A_{computed}) for a second-chance assessment is calculated as follows:

- If your first attempt is higher than your second attempt, the first attempt is used as the overall mark. If you do not take a second attempt, the first attempt is used as the overall mark.
- If the second attempt grade is higher than the first attempt grade, the final computed grade is:

$$A_{\text{computed}} = 0.75 \times A_{\text{second}} + 0.25 \times A_{\text{first}}$$

- Thus, if you do not take the first attempt, your maximum score is 75% assuming a perfect score on the second attempt.
- E.g., A score of 70% on attempt 1 and 95% on attempt 2 results in $(0.75 * 95 + 0.25 * 70) = 88.75\%$ for that assessment

5.3. Tutorials

Tutorials are designed to give you practice with the course material, learn new material to prepare you for the upcoming weeks, and provide you with an opportunity to meet and collaborate with other students and with your TAs. Tutorials can earn up to 10% of your overall grade as Engagement Tasks (see [Section 5.4](#) for more information).

While tutorial attendance is technically optional, there will be important readings and exercises released with each one that are considered required readings and may be tested.

Tutorials will begin with a Wooclap poll, and you are expected to chat with classmates and discuss the results. These questions will help you prepare for the tutorial and/or related materials. Half of the tutorial mark is based on participation in the Wooclap poll. At the end of the tutorial, the TAs will ask you questions about the tutorial exercises and you must demonstrate your understanding and participation to receive the other half of your marks. You might be required to reach a certain amount of completion to receive the mark.

If you do not meet the requirements at the end of the tutorial and did not receive your mark, if time permits, TAs can review your exercise at the beginning of the following tutorial. You can only receive the mark if you also attended the previous tutorial.

5.4. Engagement Tasks

There are many ways to earn engagement marks in this course. There are more marks available than are required and you can earn a limited amount of bonus marks for completing more than the required amount. These are designed to encourage practice and engagement with the course material, and some more may be added throughout the term. While it is optional to go beyond the minimum required amount of marks for your grade, all tasks are strongly encouraged to help you better understand the material.

Homework Exercises and Assignment Bonuses must be submitted to the **Engagement Task** private forum on Brightspace, which is only visible to you and the teaching team. Each task will be marked complete / incomplete and a mark will be assigned to each post. Attendance and mini-quiz marks are calculated automatically. The Engagement Task mark is the sum of all of these marks, capped at the weight described earlier. If you would like additional feedback on any engagement tasks that you submit, please add a comment requesting feedback on specific areas within the post.

An initial list of Engagement Tasks is available in [Appendix A](#).

Simple Path: Completing every tutorial and attending every lecture will earn the full 14% required for the course. Completing Assignment 0 OR completing all of the weekly mini-quizzes will earn you enough for the bonus marks. Missed lectures and tutorials will not receive accommodation, and attending late and missing the polls may not receive any accommodation. As such, you should plan to complete some of the additional tasks to make up for extenuating circumstances.

5.5. Accommodations and Missed Work Policies

Late Work: This course does not apply per-day late penalties. Late or missed assignments and term tests are instead handled through the Second-Chance Policy ([Section 5.2](#)) and Engagement Tasks ([Section 5.4](#)) rather than incremental deductions. For circumstances the course policies do not cover, short- and long-term academic consideration follows Carleton's process, including the self-declaration form available through the [Academic Consideration Policy](#).

Assignments, Term Tests, and Engagement Tasks: You do **not** need to request any accommodation for these assessments as the existing policies already account for missed work.

- If you are unable to complete an assignment or term test for a short-term reason, you can use the second-chance policy to still submit with a small penalty for not completing the first attempt as described in the second-chance policy.
- There are more than enough engagement tasks available to earn full marks, and bonus marks can be earned for extra credit to make up for some missed work.
- For long-term accommodation or if you are unhappy with the penalties incurred from the second-chance policy, the re-weighting policy described earlier will apply.

Final Project: The end of term Final Project must be taken seriously and can only be accommodated for serious circumstances. If you are ill or otherwise unable to complete the final project due to extenuating circumstances beyond your control, follow these steps:

1. Review the [Academic Consideration Policy](#) to understand the process and requirements.
2. Determine if your circumstances meet the extenuating criteria described and reach out to the instructor detailing how long you were incapacitated from work and the time required to complete the project.
3. If accommodations are warranted and possible, you will be asked to fill out the appropriate declaration described in the policies linked above.
4. Short-term extensions may be available, but longer-term accommodations will be handled on a case-by-case basis and may involve re-weighting and delaying your final grade until the project can be completed and assessed.

Final Exam: The final exam is scheduled by the Registrar's Office and accommodations are handled through them. Take time to review the [Deferral Policy](#) and the [Academic Consideration Policy](#) to understand the process and requirements for requesting a deferral or accommodation for the final exam.

5.6. 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, please reach out to classmates for notes and discussion to catch up.

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://wellness.carleton.ca/mental-health/resource-guide/>

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! Review the communications policy for more information about how to reach out for help with in-class materials. The following are some helpful resources for general support:

- **General academic skills support?**

- Science Student Success Centre: <https://sssc.carleton.ca/>
 - Peer mentoring, workshops, industry events.
- Carleton Computer Science Society Events: <https://ccss.carleton.ca/events/>
 - Workshops, study groups, community events, and more.
- Centre for Student Academic Support: <https://carleton.ca/csas/>
 - Time management, study skills, organization, general academic skills.

- **Technical support?**

- Virtual Machine Overview: <https://carleton.ca/scs/tech-support/virtual-machines/>
- Virtual Box Technical Support Guides: <https://carleton.ca/scs/tech-support/virtual-machines/virtual-machine-technical-support/>
- OpenStack Technical Support Guides: <https://carleton.ca/scs/tech-support/scs-open-stack/openstack-technical-support/>
- SCS Technical Support Procedures: <https://carleton.ca/scs/tech-support/contact-it-support/>

- **Less-academic support?**

- Mental and Physical Wellness: <https://carleton.ca/wellness/>
- Ombuds Services: <https://carleton.ca/ombuds/>
 - Confidential, impartial, and independent support for students
 - Non-academic misconduct, harassment, sexual violence, issues with housing/landlords, problems with courses or faculty or staff, university administration, student accounts, scholarships, etc.

6. Communication Policy

In order for the teaching team to work effectively and to maintain a healthy work-life balance, it is important to follow the communication policy outlined below to receive the most timely and effective responses to your questions and concerns. I **strongly encourage** questions and discussion, but in the appropriate places at the appropriate times.

Announcements: It is your responsibility to read the course announcements **each day**. They will not release daily, but you must keep up to date with them as it may have important or urgent information.

Lectures: Important course updates will be announced during lectures. It is your responsibility to attend or get information about any lecture announcements from a classmate if you miss a lecture.

Brightspace Discussion Forums: This is the primary place for questions and discussion. TAs will be monitoring the forums multiple times daily and are expected to respond to all questions within 48 hours on weekdays and forward any difficult questions to the instructor to review within 72 hours. The goal is to have all information that can be asked publicly to be answered publicly so that all students can benefit from the discussion.

Emails: In general, there are few reasons to email the teaching team directly and most questions should be asked on the discussion forums so that clarifications and support can be provided to all students. Emails about simple clarifications, dates and deadlines, confusion about materials, technical support, will all be redirected to the discussion forums or responded to via announcement or in-class.

- **General Inquiries:** Some emails that are common concerns or are easily answered by the course outline may be not be responded to directly and may be addressed in an announcement to the class via Brightspace or in lecture. Review announcements after sending an email to see if your question was answered there.
- **TA Emails:** You will not need to email your TAs directly.
- **Lab Coordinator:** Email regarding issues with TAs that cannot be resolved with the TA.
- **Instructor:** For accommodations, serious academic concerns, concerns about your wellbeing, or other personal matters that are not appropriate for public discussion forums, email the instructor. Make sure to include your student number and a clear subject line with the course code. It is also helpful to include a note of how you would like to be addressed (e.g., Connor, Mr. Hillen, etc.) in the email.
- **Response Time:** Emails will be responded to within three business days. Do not expect to receive responses during evenings, weekends, or holidays. If you do not receive a response within three business days, please review the announcements to see if your question was answered via an announcement to the class, and if not, please follow up with a polite reminder in the same email thread. If you still do not receive a response, reach out during student hours or before/after class to make sure the emails are being received.

Grade Disputes: If you believe that an error was made in the grading of your assignment, reach out in the Private Grade Dispute forum on Brightspace within **three (3) days** of receiving your grade. Include the name of the TA that graded you in the title. Even if the claim is valid, grades will not be reviewed if disputes are submitted after this deadline. Once the request is submitted, it may take longer than this deadline to resolve, but this is okay. If a TA does not respond within three business days, please reach out via email to the lab coordinator to follow up. **NOTE:** Technical issues, missed deadlines, disagreeing with the marking scheme, or submission mistakes are **NOT** valid disputes.

Professionalism: All communication should be respectful and professional. We will not tolerate abuse or hostility towards the teaching team or other students and any abusive or hostile communication will be reported to the Dean's Office. If you feel that you are being treated unfairly or disrespectfully, please email the instructor and lab coordinator to discuss the situation.

TA Student Hours: TA student hours TBA via Brightspace.

Instructor Student Hours: This is where you can reach out for extra support with course concepts, academic concerns, personal concerns, accommodations, or - while lower priority - chat about topics related to the course, career, or CS in general. While the hours are open for discussion, they may be prioritized by specific need and urgency. Hours will be posted to Brightspace shortly after the beginning of the term.

7. Course Scheduling and Modality

This course is being taught in-person and attendance is expected for success in the course. In-class announcements may be the only source of important information for the flow of the course. There are no recordings for this course.

Lectures may be switched to Zoom if the instructor falls too ill to attend in-person or during a major weather event. In these events, announcements will be sent out ahead of the lecture with links and relevant information. Check-in on announcements daily to keep up to date with any changes to modality. These online emergency lectures will be recorded and released.

8. Important Considerations and Expectations

Most of these considerations can be summed up as: Start early, back up your work, submit progress frequently, and follow the course policies.

1. **Technical Issues are NOT grounds to overrule policies.** 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.
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.
 - It has happened **very frequently** in this course that students have received zeroes on assignments and even the final project due to corrupt files or missing files being submitted.
 - Give yourself time to download the submission from Brightspace, put it onto the virtual machine, extract, compile, run.
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/>
 - If you would like a previous submission to be graded rather than your first submission, please contact the course lab coordinator within 24 hours of the original submission.
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**.
6. **All quizzes, tests, and exams are expected to be completed individually and without electronic aid:** any communication or access of electronic devices (e.g., cellphones) during a quiz, test, or exam will be considered cheating. Use of Smart Glasses (or other similar assistive technologies) before or during a quiz, test, or exam is strictly prohibited and is considered an academic integrity violation. Even if you have prescription glasses with assistive technologies, you will not be allowed to use them during quizzes, tests, or exams, so please make sure that you have acceptable prescription glasses. The proctor has the right to ask to inspect your glasses before or during a quiz, test, or exam. In addition, unless authorized with PMC accommodations, you are NOT allowed to use headphones nor earbuds during a quiz, test, or exam. Also, during a quiz, test, or exam, if you choose to wear a baseball cap, sunglasses, or anything else that would prevent a proctor from monitoring where your eyes are focused, you will likely be asked by a proctor to move to a different seat. Lastly, you must submit your quiz, test, or exam before leaving the classroom, or it will not be graded.

9. Plagiarism Policy

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/>.

1. General expectations for academic integrity in this course include:

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),

2. Assignments 1 - 3, Homework, Tutorial Exercises, Weekly Quizzes; You are permitted to:

1. Discuss assignments and related concepts with classmates and chat AI
2. Provide debugging support to classmates, such as helping understand error messages, using debugging tools, and providing a review of small parts of code
3. Use online forums, documentation, and generative AI to help understand concepts and generate small amounts of code. See [Section 10](#) for information about use of generative AI in this course. Note that you are still **not** permitted to share any part of the assignment specification or provided code with any online sources or chat-based AI tools.
 - Avoid having an AI generate whole functions or implementing whole algorithms. AI should primarily be used for basic syntax, support, or repetitive coding snippets, NOT for making design decisions.
4. In any case where you receive support from another source, you **must** cite that source in the comments of the code at the top of the file and in the README file submitted with each assignment.
5. All of the work that you submit **must be your own**. Simply re-ordering code from another source or changing variable names / comments is not sufficient. You must have a clear understanding of the code that you submitted **and be able to explain it if called upon to explain your work**.

3. Term Tests 1-4, Second-Chance Online Retake:

1. You **MAY** Work individually and use a non-programmable calculator to assist with calculations.
2. You may **NOT** use generative AI, course notes, compilers, or additional resources. This ensures that your grade on the test reflects your understanding well-enough for you to gauge your capabilities before the final exam.

4. The Final Project follows different rules as a final assessment of your understanding of the practical course material:

1. You are permitted to work as a pair with another classmate, but are not required to. The project is expected to be able to be completed by an individual within the provided time. You **MUST** cite your classmate in your submission and submit as a group on Brightspace. Only begin work together once you are signed up as a group on Brightspace.
2. You **are** permitted to use official documentation for the C programming language and the official course notes.
3. You **are** permitted to use past course notes, exercises, and previous assignments to assist with the project, but you must cite their usage in your README files to avoid being incorrectly flagged for misconduct.
4. You are **NOT** permitted to use online Q&A boards, finding code snippets/examples online, or generative AI to write code for your project. This includes but is not limited to GitHub Copilot and ChatGPT.
5. You are **NOT** permitted to provide debugging support to classmates who are not in your group. Debugging is a skill we want to assess.

10. Generative AI

Discussing AI Use: As our understanding of the uses of AI and its relationship to student work and academic integrity continues to evolve, students are required to discuss their use of AI in any circumstance not described in this outline with the course instructor to ensure it supports the learning goals for the course.

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 knowledgeable professionals from workplaces, or otherwise worse working conditions and bargaining capability.

From an educational perspective, there are some ways to work with the AI that can be beneficial, and many ways it can be harmful. As described earlier in [Section 2](#), it is easy to use AI to find the right answer, but in doing so, avoid the work that builds understanding of the material. The end result of using AI could mean that you end the course with a high grade, but exit with no new understanding as though you never took it at all. This is wasteful, does not give us as instructors a clear picture of whether our course is effective, and you will not be prepared for future courses or for your future career. What will set you apart from others is **real understanding** earned by working through the material. Here are a few tips for use of generative AI, noting that AI in this course is permitted under only a few circumstances described in [Section 9](#):

1. **AI is for Experts:** Generally, AI is best used when you already have a strong understanding of the material. When you first get started in a new area of understanding, you are usually not equipped with the knowledge of *how to verify* whether information is correct or not. This can lead to accepting incorrect or poor quality AI results without any understanding.
2. **Muscle Memory:** It's strange, but when you first get used to a new language you must build up a kind of mental muscle memory for the syntax. This means taking the time to write out the code, even if it is slow and inefficient. Once it is second-nature, AI can help to streamline it, but don't over-rely on it at the start.
3. **Distrust, and Verify:** Do not assume that the AI results are correct. Do not move forward until you know what each term and line of code does. If you do not understand, find external sources to verify the information. Do not accept results without being confident in your understanding.
4. **Get Verifiable Support:** In areas that can be immediately verified, AI can be very helpful. For example, if you aren't sure how to navigate the Linux terminal, an AI can help you with the right commands. It is instantly verifiable, because it will either work or not work. Then you can dig deeper into the documentation to understand why it works.
5. **Use AI for Exercises:** Ask for hints or guidance, explicitly ask to not receive a full solution.
6. **Clarify Readings:** Ask for clarification about a reading in the course notes and how topics might relate to each other, knowing that the AI might be misleading. "I am struggling with my course reading about X, but I might be misunderstanding something from earlier in the course. Can you quiz me on a few essential topics required to properly understand this material to see where the gap in my prerequisite understanding is?"
7. **Use AI to Help Learn the Skill of Debugging:** Do not just have AI write code for you or debug for you. Use it to help learn the skill of debugging. For example:
 1. "I'm trying to improve my debugging skills without being given the answers outright. How do I read this error message that I'm getting when I compile my C code?" - don't even provide it the code to debug.
 2. "Where should I be looking in my code to find the source of the error, and what about this error message would lead me to that part of the code in the first place?"

11. School of Computer Science Information

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/>.

12. 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](#).

Appendix A: Engagement Tasks

These engagement tasks may be updated and adjusted as the term progresses. Review and in-class discussion for the most up-to-date information. **Bolded** tasks require a submission to the Private Engagement Task Forum on Brightspace.

The marks below are based on OVERALL grade. E.g., attending 7 lectures at 0.2% each means that you earn 1.4% of the 14% required. Bonus tasks can earn up to 2.5% above the base, and your total Engagement mark is capped at 14% (16.5% with bonus).

TASK	MARKS EACH	MAX	NOTES
Assignment 0	2.5%	2.5%	Completion of Assignment 0, a self-study assignment that onboards you to the C language during the first two weeks. Must be submitted to the Engagement Task forum before Assignment 1 releases, but can be resubmitted if an UNSAT is received. Support will be given during the first two tutorials.
Lecture Attendance	0.2%	4%	Based on participation in Wooclap polls during lectures, capped at 20 lectures. If any lectures are cancelled, the weight of each lecture will be adjusted to balance to 4%/20 lectures.
Tutorial Attendance	1% / lab	10%	0.5% is received for participating in a Wooclap poll at the beginning of the tutorial and 0.5% is received after being marked by a TA at the end of the tutorial.
Weekly Mini-Quizzes	0.25% / quiz	2.5%	Quizzes will open each week covering material from the previous week and possibly readings for the week ahead. They can be taken repeatedly and the mark for each quiz is the highest attempt mark.
Homework Exercises	0.5%	Varies	Homework exercises explicitly labelled as Engagement Tasks may be released throughout the term to offer practice with the material, marked as complete / incomplete. The exact number will vary; these count toward the overall Engagement cap.
Assignment Bonuses	1% / bonus	6%+	Assignments 1–3 will include at least 6% worth of bonus tasks that can be completed during OR after the assignment submission period; more may be added. Assignment Bonuses must be submitted to the Engagement Task forum.
Project Bonuses	1% / bonus	3%+	The project will include at minimum 3% worth of engagement task bonus tasks, such as using Git (or other version control) and submitting an early planning document within the first week of the project, among others; more may be added.

The values in the MAX column marked with “+” are guaranteed minimums offered; additional bonus tasks may be added as the term progresses. Regardless of how much bonus work you complete, your overall Engagement mark is capped at 14% (16.5% including bonus).

Additional tasks might be posted to Brightspace as the term progresses, but only the above are guaranteed to be available.