

# Introduction to Systems Programming

COMP2401 A/B/C

Fall 2024

## Course Outline

<b>Instructor:</b>	Connor Hillen (He/Him)	<b>Office Hours:</b>	Posted to Brightspace
<b>Email:</b>	<a href="mailto:connorhillen@cunet.carleton.ca">connorhillen@cunet.carleton.ca</a>	<b>Website:</b>	<a href="https://brightspace.carleton.ca">https://brightspace.carleton.ca</a>
<b>Lecture Hours:</b>	<b>A:</b> Mon. Wed. 10:35–11:55 <b>B:</b> Tue. Thu. 16:05–17:25 <b>C:</b> Wed. Fri. 14:35–15:55	<b>Location:</b>	Full Details Posted to Brightspace <b>A:</b> In-Person <b>B:</b> Online (Zoom) <b>C:</b> Online (Zoom)

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

## 1. About the 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 synchronisation, inter-process communication, file systems, networking, pointers, heap and stack memory management, and system/library calls.

**Prerequisite(s):** (COMP 1006 or COMP 1406 or SYSC 2004) with a minimum grade of C-.

**Textbooks and Other Resources:** In lieu of a textbook, course notes have been made available online.

- M. Lanthier, C. Laurendeau, and D. Nussbaum, *COMP2401 Course Notes*. [Online, Brightspace]

**Objectives:** This course is here to introduce you to the C programming language and the introductory concepts of systems programming which underlie your computer programming and computer use. The objective is to introduce you to the principles behind systems programming, gain some proficiency working in the C programming language and building C programs, and be comfortable writing and designing programs which work at a lower level and necessitate working with memory directly.

**Topics Covered:** While minor areas may be adjusted for time, students who complete all material, attend all lectures, and perform all readings 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,
- Organise program structure to follow clean coding practices in C,
- Perform file input/output tasks,
- Implement concurrent programming techniques, including process management and threads,

**Technology:** All students are required to work using the course virtual machine (VM). Information about the virtual machine will be posted to Brightspace. Make sure to use the latest VM for COMP2401. Learn more about VMs here:

<https://carleton.ca/scs/tech-support/virtual-machines/>.

Online lectures and online tutorials will be held via Zoom (requires Carleton login) and communication will be primarily handled via Carleton Email and Brightspace Forums. Zoom links will be available on Brightspace.

## 2. Teaching Statement

---

Welcome to COMP2401! You can refer to me as “Connor” or “Mr. Hillen”, whichever you are most comfortable with. I hope you are excited to look at programming in a new way than you might have looked at it in previous courses. This course can be challenging, but I am confident that with the appropriate effort, **all** students are capable of succeeding at reaching our course outcomes. When reaching out, I invite you to include your preferred name and/or pronoun.

Learning happens best when people actively engage in material on a regular basis. This means you will need to engage in reading, practicing, recalling, synthesizing, and analyzing the material in various ways, consistently. I have organized this course to provide opportunities to practice and receive feedback, to combine learning in-lecture with self-paced learning through readings and writing programs, and to include opportunities to improve marks for some learning objectives after receiving feedback (see: Section 3).

We try to reduce unintentional bias and inconsistencies in grading by using strict rubrics and automated grading where possible. Mistakes can happen, and there is always room for improvement, and this is why I am a believer in receiving and acting on feedback. Throughout the term, you can reach out to me to discuss issues you are having with the course, to let me know if any material or behaviours have made you feel uncomfortable, and there will be multiple, anonymous surveys to provide feedback to me for ways to improve this course and my teaching of it.

I **expect honesty** from both the students and from my teaching team. From students, I want honest demonstrations of *your* ability on tests and assignments. This helps me to know where my teaching needs improvement and that you really are prepared for the next term. From my team, we will provide honest feedback and communicate openly about our expectations for the course. I will admit mistakes and gaps in knowledge and work to improve.

I **expect respect and responsibility** from both the students and from my teaching team. Students should be respect each other’s viewpoints and abilities, and take responsibility for showing up, doing the preparations, being physically and mentally present during instruction, and seeking extra support when needed. As the teaching team, we will show the same respect of viewpoints and abilities while making sure that we challenge deeper and more critical thought. We will be responsible for providing timely feedback and organizing the course in a way that both supports your learning and makes sure that you leave meeting our standards.

I expect us to have **fairness and trust** between each other. As students, do not try to freeload off of partners or peers, contribute fully to any collaborative work, do not seek an unfair advantage over your peers, and use class time for class time. Our teaching team is committed to treating students equitably, to follow through on our promises, to be available when we say we will, and to avoid changing expectations without clearly communicating this in a timely way.

Finally, we all need a bit of **courage**. I accept that upholding these values and structuring my courses in this way, though it is informed by good practice, may lead to poor reviews by some students or other consequences, and that it can be uncomfortable to uphold these values at times. Similarly, students should have the courage to accept lower or even failing grades and other consequences to uphold our values and follow policies. When you see these values being broken, even by me - the instructor - please say something. I want to improve and to make sure we have an environment that is conducive to learning.

I ask that you see this course as more than just a grade, but recognizing that really understanding this course can be a major asset to supporting your future coursework, your career, and your understanding of programming and computer science as a whole! I will do my best to be available to help keep you on track and to provide what information I can to motivate success in the course, but I need you to do your part, too. Take advantage of the legitimate resources available, trust in the structure of the course, engage in the materials provided, and give me feedback with what is working and what is not working. And **please** read the rest of the syllabus carefully!

## 3. Assessments

---

**Grading:** To ensure students have an opportunity to practice the material before receiving a final grade, there are several components which are low or no weight and intended to primarily give you practice and feedback, while the larger summative items are a final gauge of your understanding of the material. You should aim to complete all of the materials in the course to be sufficiently prepared for the summative assessments. Some grades are also allocated to “Side Quests”, a list of marking items where you choose which ones to complete from the provided list, designed to engage you in different parts of the course goals that you find most interesting, reward consistent participation, and support greater curriculum goals. You are *expected* to participate in all course elements to be fully prepared for the summative elements.

**Note:** All sections of this course must participate in an in-person final exam. You can receive up to **1.5%** bonus marks for completing additional side quests beyond the required amount listed below.

Assignments (6x1%) .....	6.0%
Projects (2x17%) .....	34.0%
Unit Tests (2x7.5%) .....	15.0%
Final Exam .....	40.0%
Side Quests .....	5.0%

### 3.1. Tests, Quizzes, Exams

**Unit Tests:** There will be **two Unit Tests** which approximately cover our major course units. These tests are held online, asynchronously via Brightspace, and must be completed individually. These tests provide an opportunity to identify areas of weakness before the final exam.

**Summative Grade Override:** To promote learning from feedback, a higher grade on the relevant sections of the written final exam will replace the relevant unit test marks. You will not receive feedback on your final exam unless requested, so use the unit tests to find areas of weakness to focus on for study and receiving assistance.

- **Unit Test 1:** 09:00, Tue. Oct. 15 – 18:00, Wed. Oct. 16
  - Topics Subject to Change: Binary, base conversions, data types, C user I/O, compound data types, pointers
- **Unit Test 2:** 09:00, Tue. Oct. 15 – 18:00, Wed. Oct. 16
  - Topics Subject to Change: Dynamic memory, concurrency, processes, multithreading, building, sockets, signals, buffers

**Weekly Progress Check Mini-Quizzes:** Every Friday, there will be weekly “Progress Check Mini-Quizzes” held on Brightspace to provide quick feedback on your understanding of the week’s materials. You will be given multiple attempts to complete them, but they must be completed within one-week of their release. These quizzes are ungraded, but consistent completion of them can provide marks toward your “Side Quest” mark component, and will provide instant feedback for areas of improvement each week.

**In-Class Polls:** Lectures will frequently include engagement polls to check-in on understanding of the material. These polls are not graded, but consistent completion of them can provide marks toward your “Side Quest” mark and help drive the lectures in the necessary direction to support everyone.

**Final Exam:** The Final Exam is scheduled by the registrar and will be **closed book, in-person** and will cover materials from both unit tests and the mini-quiz material which came after the unit test materials.

### 3.2. Assignments, Tutorials, Projects

**Projects:** The two course projects are intended to provide a final assessment of the applied learning outcomes. You will **not** receive extensive feedback on your projects, as they are intended to assess your final understanding of the topics, except where specified in the project specifications. Students may be permitted to work in pairs for the projects; details and some restrictions will be discussed in class and on the project specifications.

- **Project 1:** Available (Latest) 00:00, Wed. Oct. 09 – 23:59, Wed. Oct. 30
  - Topics Subject to Change: Binary, user I/O, arrays, structures, strings, pointers, code design
- **Project 2:** 00:00, Wed. Nov. 06 – 23:59, Wed. Nov. 27
  - Topics Subject to Change: Dynamic memory, multithreading, building, makefiles, code design

**Assignments:** Assignments run almost each week, and for approximately one week each. The assignments are primarily given to provide practice on the materials to support your project works and are opportunities for feedback to help you learn where you need additional support. You are expected to complete each assignment in order to prepare for the projects and receive support during tutorials.

- **Assignment 1:** Available (Latest): 23:59, Sun. Sep. 15 – **Due:** 23:59, Sun. Sep. 22
  - Topic Subject To Change: **Binary and I/O**
- **Assignment 2:** Available (Latest): 23:59, Sun. Sep. 22 – **Due:** 23:59, Sun. Sep. 29
  - Topic Subject To Change: **Arrays and Pass-By-Reference**
- **Assignment 3:** Available (Latest): 23:59, Sun. Sep. 29 – **Due:** 23:59, Sun. Oct. 06
  - Topic Subject To Change: **Strings and Structures**
- **Assignment 4:** Available (Latest): 23:59, Sun. Oct. 06 – **Due:** 23:59, Sun. Oct. 13
  - Topic Subject To Change: **Dynamic Memory**
- **Assignment 5:** Available (Latest): 23:59, Sun. Oct. 27 – **Due:** 23:59, Sun. Nov. 03
  - Topic Subject To Change: **Makefiles and Complex Ownership**
- **Assignment 6:** Available (Latest): 23:59, Sun. Nov. 03 – **Due:** 23:59, Sun. Nov. 10
  - Topic Subject To Change: **Multithreading**

**Summative Grade Override:** To promote learning from feedback, a higher grade on the relevant sections of the projects will replace the relevant assignment marks. You will not receive extensive feedback on your projects unless requested, and are expected to use the assignments as opportunities to prepare for the projects.

**Tutorials / Labs:** Tutorials begin the week of September 09. Each week, there will be a weekly tutorial where you will be able to work through practice exercises, discuss the assignments and common questions with TAs and peers, and learn supporting techniques and tools to assist you with your other course work. The first few tutorials are the primary place to learn about the course technology and get started in C programming. While tutorials are not graded, participating in tutorials can provide marks toward your “Side Quest” mark and are an opportunity to discuss the material with fellow students and the teaching assistants. The targeted exercises are designed to support your applied knowledge of the materials in ways that we might not cover in class.

**Late Submissions:** Projects and Assignments will receive a 5% deduction **per hour** (rounded down) for each hour past the posted deadline on Brightspace. You have **24 hours** from the submission deadline to email the instructor and request that an earlier submission of the material be graded instead of the final submission.

**Regret Clause:** If you believe that you have committed academic misconduct while completing an assignment and regret this decision, you may email the instructor within 48-hours of the submission deadline to receive a mark of zero on the assignment. You will not face any sanctions for this case, although other parties involved will still be forwarded to the dean. The case of will not be forwarded to the dean unless a future case of misconduct is suspected.

### 3.3. Side Quests

A portion of your mark is dedicated to completing “Side Quests”, various tasks in the course related to engaging in the course, gaining extra practice with course outcomes, or engaging in work that is closely related to the outcomes or curriculum-level goals.

Some *Quests* can be completed more than once, and some might be able to be completed in pairs. Each Quest will specify the requirements. Mostly, the quests will be graded via Forum posts on the course web page. There is a private forum that only you and the teaching team can see. You can post the results of your quest there in an easy to view format and your TAs will periodically award marks on a SAT/UNSAT basis for completing the quest. To ensure quick grading, please try to post in a way which clearly demonstrates your completion of the quest. Make sure to include the “ID” of the quest in the title of your post.

Unless explicitly stated in the Quest information, each Quest must be completed individually and without using external sources like AI or external websites.

If you have an interesting idea that you think would qualify for a side quest, please reach out to the instructor. Interesting projects that are related to the course goals might be considered for marks! All of the available quests are available on the course web page, but a starting selection is available here:

ID	Name	Requirements	Value	Times Completable	Deadline
HELLO-025	Say Hello!	Post to the “Say Hello!” forum on Brightspace following the guide on the pinned post and chat with your fellow classmates.	0.25%	Once	Sep. 22nd
ASSIGN-025	Assignment Bonus	Each assignment includes a bonus section. Do not submit the completed code to the assignment slot unless specified; instead, post the completed code to the quest forum and follow any requirements on the assignment for posting.	0.25%	1 / Assign.	One Week Past Assignment Deadline
TUT-1	Tutorial	Participate in three tutorials and complete the attendance quizzes on-time. Additional submissions to side quest forum are not required. Grade may be revoked if TA flags as not participating.	1%	3 Times (9 tutorials total)	In-Tutorial
WKLY-05	Weekly Quizzes	Receive a SAT on three Weekly Progress Check Mini-Quizzes. Additional submissions not required.	0.5%	3 Times	Posted Quiz Deadlines
POLLS-05	In-Class Polls	Respond to polls during a lecture. Number of polls can vary through the term. 0.5% for attending 80% of lectures and responding to most of the polls when polled. No additional post on quest forum needed.	0.5%	Twice, once before/after Fall Break	Fall Break / End-of-Term
PLAN-1	Project Plan	Within the first week of the project, submit a basic plan for your code, including a list of tasks to complete and an outline of what code you will need to write and how it will flow. Submit a text file or PDF of your plan to the quest forum.	1%	Twice, 1/ project	One Week after Project Release
GIT-05	Git Project	Use Git or similar version control system during your project in an effective way (i.e., making multiple commits throughout the project). Submit a screenshot of your git history showing your commits.	0.5%	Twice, 1/ project	One Week Following Project Deadline



### **3.4. Support and Assessment Accommodations**

**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/>

**Unit Tests:** If you are unable to write the unit tests within the specified time, the mark will automatically be replaced by the final exam mark and accommodations do not need to be requested. If you would like to still write the test to receive feedback, please reach out to the instructor.

**Assignments:** If assignments can not be completed before the deadline, you may still receive feedback during office hours. The mark will be replaced by the relevant project marks, and accommodations do not need to be requested.

**Projects:** You are expected to start your projects *early* and submit progress on your projects *frequently*.

- **Short Term Accommodation (Less than 5 days):** Our projects are several weeks long and should **not** be started right at the deadline. Of course, some circumstances can still prevent finishing touches and final submission. If you experience a short-term incapacitation, **no later than 24-hours** after the project deadline, email the instructor with your current progress on the project and submit to the “Academic Consideration for Course Work” form here: <https://carleton.ca/registrar/academic-consideration-coursework-form/>
- **Long-Term Accommodation (Greater than 5 days):** If you are experiencing longer-term circumstances, you will need to reach out to the Registrar’s Office for support, but please email the instructor for immediate guidance when you are able.
- **Expected Accommodations:** To minimize unintentional bias, accommodations will be applied consistently where possible. You may be requested to submit a fully completed project at a later date and have the weight shifted to other course elements. Graded extensions may be considered if sufficient notice is provided before the due-date and an appropriate amount of work toward the project has been completed already, but this is not guaranteed.

**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/>

## 4. 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 they that 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**.

## 5. Course Scheduling and Modality

---

**Section A:** COMP2401 A will be held using in-person lectures that will **not** be recorded, but materials of other sections will be made available. Space permitting, students from other sections can attend in-person lectures, but priority must be given to students registered in section A. Tutorials for Section A will also be held in-person; space permitting, some students from other sections may be permitted to attend in-person tutorials.

**Section B, C:** Lectures for COMP2401 B and C will be held online via Zoom, link posted to Brightspace. These lectures will be recorded, but 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.

**A microphone is required for participation in group discussions.** A webcam, turned on, is **very strongly encouraged**. Note that I will be attempting to keep lectures in **Focus Mode**, meaning that even with a camera on, your peers can only see your camera in Breakout rooms or if you are “Spotlighted” by the instructor. This helps to maintain attention, give the instructor an idea of how the information is being understood, allow the instructor to get to know the class better, all while avoiding potentially distracting video feeds from other students. (I really appreciate it!)

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.

## 6. Communication Policy

---

### 6.1. Where to Seek Help

#### 1. My assignment was graded incorrectly!

- Teaching Assistant → Lab Coordinator → Instructor
- Contact the teaching assistant that graded you. Their name should be in the feedback. **You have 7 days** following the receipt of your grades to send the email to dispute the grade with your teaching assistant. We will try to resolve the matter **within 5 days of receiving the email**. If you are missing a grade, missing the name of your grader, or cannot resolve the issue, please contact our lab instructor.

#### 2. I don't understand material!

- Review lectures → Course Notes → Forums / Teaching Assistant Office Hours → Instructor
- Check out Section 3.4 for more information about getting support. Remember, we're here to help! Ask for help when you have tried to understand and reviewed the provided materials but can't quite figure it out.
- Asking questions on forums can be very helpful, as it can help multiple students experiencing the same issue, but sometimes you need someone to talk through it with you.

#### 3. I need help debugging!

- Slow Down → Review Specification → Carefully Read Your Code → Teaching Assistant → Instructor
- **You will have difficulty debugging** in this course. It is a vital skill to learn. The majority of office hours that are used for debugging will often come down to the following steps:
  1. If there is an error message, carefully **read and understand** the error message. Exactly what is it telling you?
  2. List what **could** be wrong based on the bad behaviour. What could **possibly** result in this behaviour?
  3. Narrow it down. Comment out parts of the code until you isolate where the problem behaviour is.
  4. **Very, VERY carefully actively read your code**. Do not make any assumptions. Follow your code on paper keyword for keyword, starting well before the problem emerges. Consider each line extremely carefully, as even a single symbol out of place can cause very unexpected behaviour.
  5. Plan ahead next time. Do **not** just start writing code. Plan it out and test as you go. Move on to new functionality only when you are confident.

#### 4. Tech Issues: Laptop, Virtual Machine, OpenStack:

- Check Forums → Check [SCS Technical Support Documents](#) → Post to Forums → TA Office Hours → Lab Coordinator (if TA told you) → SCS Tech Support

#### 5. Private / Personal Concerns: For any private or personal concerns, please contact the Instructor via email.

### 6.2. Conduct and Expectations

1. Expect responses **within three business days** during the hours of 08:30 - 17:30. To help maintain a school-work-life balance, the teaching team has been instructed not to respond on weekends or evenings by the instructor.
2. Do not post solutions to forums. A single line of code is typically okay, but do not outright post solutions or problematic code so that others can go through the learning process. If code must be shown, discuss it with a TA.
3. **Email should be for private, confidential communication only.**
4. Emails must:
  1. Include "COMP2401" in the subject line
  2. Come from your official Carleton email
  3. Include your name and student ID (ideally preferred name / pronoun to address you)
5. Be kind, be understanding, be respectful, and we will commit to doing the same. Report bad behaviours from students or TAs to the instructor and lab coordinator as soon as possible.



## 7. Plagiarism Policy

---

### 7.1. 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. Grades are a useful tool for the teaching team to know if our learning goals are being achieved, and help guide our decisions for improving the course.

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. Your peers will unfairly receive lower grades despite putting in the work. You may be less likely to ask for the help you need to really learn and improve. 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, even when it is challenging. If you are struggling with material, or time, or don't understand the purpose of the materials, please reach out to me.

### 7.2. 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 8 for information about *why* these policies are in place.

## 8. 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 hinder the learning experience, even unintentionally. 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.

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 7.2), 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!
  - 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.

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.

## 9. School of Computer Science Information

---

### 9.1. SCS 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/>.

### 9.2. 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](mailto: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.

### 9.3. 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/>.

## 10. University Policies

---

### 10.1. Academic Accommodations

Carleton is committed to providing academic accessibility for all individuals. Please review the academic accommodation available to students here: <https://students.carleton.ca/course-outline/>.

### 10.2. Academic Integrity

**Student Academic Integrity Policy.** Every student should be familiar with the Carleton University Student Academic Integrity policy. A student found in violation of academic integrity standards may be sanctioned with penalties which range from a reprimand to receiving a grade of F in the course, or even being suspended or expelled from the University. Examples of punishable offences include plagiarism and unauthorized collaboration. Any such reported offences will be reviewed by the office of the Dean of Science. More information on this policy may be found on the ODS Academic Integrity page: [Academic Integrity | Faculty of Science \(carleton.ca\)](#).

**Plagiarism.** As defined by Senate, “plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one’s own”. Such reported offences will be reviewed by the office of the Dean of Science. More information and standard sanction guidelines can be found here: <https://science.carleton.ca/students/academic-integrity/>. Please note that content generated by an unauthorized A.I.-based tool is considered plagiarized material.

**Unauthorized Collaboration.** Senate policy states that “to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis”.

# 11. Preliminary, Tentative Course Calendar

This is a tentative calendar, subject to change. Please report any inconsistencies of deadlines between the course website and calendar to the instructor. Remember that all dates are subject to change; review your Carleton email daily to keep up to date. A calendar with full lecture topic descriptions is also available on Brightspace. Check Carleton Central for information about your tutorial scheduling.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
<b>Week 1</b>	01 Sep.	02	03	04	05	06	07
<b>Week 2</b> T1: Linux and VM	08	09	10	11	12	13	14
						<b>Section C Cancelled</b>	
<b>Week 3</b> T2: Warnings and Errors	15	16	17	18	19	20	21
	<b>Assignment 1 Released</b>						
<b>Week 4</b> T3: Bits, Reviewing A1	22	23	24	25	26	27	28
	<b>Assignment 2 Released</b> <b>Assignment 1 Due</b>						
<b>Week 5</b> T4: Strings, Reviewing A2	29	30	01 Oct.	02	03	04	05
	<b>Assignment 3 Released</b> <b>Assignment 2 Due</b>						
<b>Week 6</b> T5: Heap, Reviewing A3	06	07	08	09	10	11	12
	<b>Assignment 4 Released</b> <b>Assignment 3 Due</b>			<b>Project 1 Released</b>			
<b>Week 7</b> <b>No Tutorials</b>	13	14	15	16	17	18	19
	<b>Assignment 4 Due</b>	<b>Cancelled: Holiday</b>	<b>Unit Test 1 Opens @ 09:00</b>	<b>Unit Test 1 Closes @ 18:00</b>	<b>Section B Cancelled</b>	<b>Section C Cancelled</b>	

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
<b>Week 8</b>	20	21	22	23	24	25	26
<b>No Tutorials / Office Hours</b>		<b>Fall Break</b> No Classes	<b>Fall Break</b> No Classes	<b>Fall Break</b> No Classes	<b>Fall Break</b> No Classes	<b>Fall Break</b> No Classes	
<b>Week 9</b>	27	28	29	30	31	01 Nov.	02
T6: Debugging, Reviewing A4	<b>Assignment 5</b> Released			<b>Project 1 Due</b>			
<b>Week 10</b>	03	04	05	06	07	08	09
T7: Threads, Reviewing A5	<b>Assignment 6</b> Released <b>Assignment 5</b> Due			<b>Project 2</b> Released			
<b>Week 11</b>	10	11	12	13	14	15	16
T8: Processes, Reviewing A6	<b>Assignment 6</b> Due						
<b>Week 12</b>	17	18	19	20	21	22	23
T9: File I/O			<b>Unit Test 2</b> Opens @ 09:00	<b>Unit Test 2</b> Closes @ 18:00			
<b>Week 13</b>	24	25	26	27	28	29	30
<b>No Tutorials</b>				<b>Project 2 Due</b>			
<b>Week 14</b>	01 Dec.	02	03	04	05	06	07
<b>No Tutorials / TA Office Hours</b>				<b>Section A</b> Cancelled	<b>Section B</b> Cancelled	<b>Section A</b> Cancelled	