COMP2401 A (Winter 2026)

Introduction to Systems Programming

Land acknowledgement:

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

Teaching Team:

- **Instructor:** Connor Hillen (Lecturer, 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') 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. Tutorials are the primary time for students to receive TA support.

Course Information Lecture attendance is required for up-to-date information.

- **Lecture Times:** Mon. Wed. 13:05 - 14:25
- **Lecture Locations:** In-Person, refer to your schedule
- **Tutorials:**
  - **Start:** Begin Week 2
  - **Location:** Refer to your schedule.
  - **Times:**
    - **A1:** Mon. 14:35 – 15:55
    - **A2:** Fri. 11:35 – 12:55
  - **Attendance:** May attend other sections if space permits.
  - **Cancellations:** There will be no tutorials Weeks 1, 7, 14
- **Course Website:** https://brightspace.carleton.ca

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

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-.

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] Grading Scheme
- [Section 13] Tentative Course Calendar
- [Section 5.4] Accommodations and Missed Work Policies
- [Section 5.3] Engagement Task List
- [Section 5.5] Technical and Other Support Resources
- [Section 6] Communication Policy, Grade Disputes, & Receiving Help
- [Section 7] Important Considerations, Additional Policies, and Expectations
- [Section 9] Plagiarism and Academic Integrity Policy
- [Section 10] AI and Generative Tools Information

2. 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.
- **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.
  - **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 in the SCS computer 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 can help to get started if you encounter issues.

3. 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. Challenge is a vital part of learning.

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**. Knowing *where* to find information will not build the same vital connections in your brain than *understanding*.

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 just 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, look at the problem from a new angle. 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.

4. Topics Covered and Learning Outcomes

A full calendar with week-by-week topics and deadlines is posted on Brightspace and in Section 13.

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
   - Data types, areas of program memory, binary and hexadecimal representations, memory addresses,virtual memory
2. General Execution of Individual Processes
   - Process management, control flows and threading, basics of process scheduling, communicating between processes
3. How the C Language Translates into Executable Code
   - Compilation, linking, libraries, system calls

5. Assessment Scheme

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

ASSESSMENT	WEIGHT	DUE DATE	TOPICS
Assignment 1	5%	Jan. 25, 23:59	Chapter 1: C arrays, printing, basic I/O
Assignment 2	5%	Feb. 08, 23:59	Chapter 2, 3: Structs, strings, pointers
Assignment 3	5%	Feb. 22, 23:59	Chapter 2: Bitwise operations
Assignment 4	5%	Mar. 08, 23:59	Chapter 3: Dynamic memory
Final Project	25%	Apr. 05, 23:59	Chapter 1 - 6
Midterm Quiz 1 (Online)	5%	Feb. 04, 08:00 - Feb. 05, 20:00	Chapters 1.1 - 3.3
Midterm Quiz 2 (Online)	5%	Mar. 18, 08:00 - Mar. 19, 20:00	Chapters 3.3 - 6
Final Exam	35%	Scheduled by Registrar	Chapters 1 - 7
Engagement Tasks	10% (2.5% bonus)	Various, latest submission: Apr. 05	Practice

5.1. Second-Chance Policy

To promote learning through mistakes and feedback, this course will be permitting the re-submission of Assignments 1 - 4 and re-taking of Midterm Quizzes 1 and 2 with the following conditions:

- **Assignments 1 - 4:**
  - May be re-submitted and re-assessed within **one week** of when grades are released,
  - 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 submission **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.
- **Midterm Quizzes 1 and 2:**
  - May be re-taken **one week** after the original quiz is released,
  - Class will be cancelled for the first attempt but will not be cancelled for the second attempt,
  - The overall mark for each quiz uses the “Weighted Average with Insurance” scheme described below.
- Additional requirements to qualify for a re-submission may be added during the term and announced in advance of the first attempt deadline, if they arise
- Due to limitations, Brightspace may not accurately reflect the combined mark for an assessment
- 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.**

Weighted Average with Insurance: The mark for a second-chance assessment is calculated as follows:

- The overall mark for an assessment, A_o, combines the first attempt, A_1, and the second attempt, A_2, as follows:

$$A_o = \max(A_1, 0.85 \times \max(A_1, A_2) + 0.15 \times \min(A_1, A_2))$$

- 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.
- Otherwise, the overall mark is 85% of the highest mark of the two attempts and 15% of the lowest of the two attempts.
- Thus, if you do not take the first attempt, your maximum score is 85% assuming a perfect score on the second attempt.
- E.g., A score of 70% on attempt 1 and 95% on attempt 2 results in $(0.85 * 95 + 0.15 * 70) = 91.25\%$ for that assessment

5.2. Re-weighting Policy

- A higher score on the Final Project will automatically replace any lower assignment scores,
- A higher score on the Final Exam will automatically replace any lower Midterm Quiz scores,
- **Note:** Not completing an assignment in anticipation of this policy is NOT recommended as this effectively increases the weight of the project and leaves your grade more vulnerable to issues encountered at the end of the term. Make use of the second-chance policy and allotted time to complete your work for practice, feedback, and better grade security.

5.3. 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 about 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 optional, they are strongly encouraged even if you meet the grade requirement.

Due to the nature of these tasks, the grade may not be clearly visible on Brightspace.

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.

TASK	MARKS EACH	MAX	NOTES
Lecture Attendance	2.5% / half of semester	5%	Based on participation in in-class Wooclap polls. Up to 5% available, once before the break and once after. Must attend approximately 80% of lectures before the break and 80% after the break to receive full marks. Part marks not available.
Tutorial Attendance	0.5% / lab	5.5%	Based on submission of a password to Brightspace quizzes. Password is given by the TA during each tutorial.
Weekly Mini-Quizzes	0.5% / mini-quiz	5.5%	Quizzes will open each week covering material from the previous week and possibly readings for the week ahead.
Homework Exercises	0.5% / problem set	Unknown	Homework exercises may be released throughout the term to offer practice with the material, marked as complete / incomplete. The exact number will vary.
Assignment Bonuses	1% / bonus	4%	Some assignments may have bonus questions that can be completed for extra engagement marks.
Project Bonuses	1% / bonus	2%	The project will include at minimum 2% worth of engagement task bonus tasks.

While there are multiple ways to earn the marks, full lecture attendance, tutorial attendance, and completion of all mini-quizzes will earn full engagement marks and bonus marks with room for error.

5.4. Accommodations and Missed Work Policies

Assignments, Midterm Quizzes, 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 midterm quiz 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.

5.5. 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:** For emails sent to the TAs, please CC the lab coordinator. Note that most inquiries to TAs should be addressed on the appropriate discussion forums.
- **Lab Coordinator:** Email regarding issues with TAs that cannot be resolved with the TA directly.
- **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 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 Review forum on Brightspace within **one week** of receiving your grade. Even if the claim is valid, grades will not be reviewed if submitted after this deadline. Once the request is submitted, it may take longer than the deadline to resolve, but this is okay. If the 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: Our primary TA student hours will occur during tutorials. Additional hours may be posted on Brightspace and announced in class throughout the term as needed. These are times to discuss exercises, homework, technical support, assignment support, and content misunderstandings. It is **not** a time for the TA to sit down and debug your code, but they can help learn how to debug for yourself.

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.

7. 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**.

8. 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.

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/.

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),

Specific policies for this course are as follows:

1. **Assignments 1 - 4, Homework, Tutorial Exercises, Weekly Quizzes**; You are permitted to:
    1. Discuss assignments and related concepts with classmates
    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.
    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 asked.
2. **Midterm Quizzes 1 and 2**; You are permitted to:
    1. Work individually and use a non-programmable calculator to assist with calculations.

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

1. You are permitted to discuss the project with one other classmate, but you are **NOT** permitted to share or review each other's code and you both must cite each other in your README files.
2. You **are** permitted to use official documentation for the C programming language and the official course notes to assist with the project.
3. You **are** permitted to use past course notes, exercises, and previous assignments to assist with the project, but you must note these sources in your README files.
4. You are **NOT** permitted to use online resources or generative AI to assist with your project. This includes but is not limited to GitHub Copilot and ChatGPT.
5. You are **NOT** permitted to provide debugging support to classmates or to receive debugging support from classmates. Debugging is a skill we want to assess.

10. Generative AI

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

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

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 3, 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. As such, it is more difficult to verify output from AI.
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. **Use AI to Dig Deeper:** Use AI to identify related topics and useful resources to learn deeper. For example, you may be struggling with pointers. Keep asking "How" and "Why" questions, and then ask for resources that can help you to verify the information once you understand a bit better. Always remember that the AI might be making up information.
7. **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. Ask for an analogy, to make a concept map, or to prompt you with questions to gauge your understanding of pre-requisite topics.
8. **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.

13. Approximate Course Calendar for Section B

The following is an **approximate** calendar and the dates and exact topics are all subject to change. Keep an eye on announcements in-lass and on Brightspace for any modifications to this schedule.

Week 1 (Jan 04 - Jan 10)
- Tutorial: No Tutorials this Week
- Mon. Jan 05: Lecture 1 Topic: Instructions, Addresses, and Compiled Programs
- Wed. Jan 07: Lecture 2 Topic: Pointers and Addresses
- Thu. Jan 08, 08:00: **Assignment 1** (Intro to C and Linux) Releases

Week 2 (Jan 11 - Jan 17)
- Tutorial 1: Welcome to Linux
- Mon. Jan 12: Lecture 3 Topic: Numerical Base Conversions
- Wed. Jan 14: Lecture 4 Topic: Data Types and Bit Models

Week 3 (Jan 18 - Jan 24)
- Tutorial 2: Debugging
- Mon. Jan 19: Lecture 5 Topic: Structs and Unions
- Wed. Jan 21: Lecture 6 Topic: Strings

Week 4 (Jan 25 - Jan 31)
- Tutorial 3: A2 & Approaching Assignments
- Sun. Jan 25, 08:00: **Assignment 2** (Pointers and Arrays) Releases
- Sun. Jan 25, 23:59: **Assignment 1** due
- Mon. Jan 26: Lecture 7 Topic: Bitwise Operators
- Wed. Jan 28: Lecture 8 Topic: Reviewing Strings, Pointers, Areass of Memory

Week 5 (Feb 01 - Feb 07)
- Tutorial 4: Strings & Structs
- Mon. Feb 02: Lecture 9 Topic: Dynamic Memory Allocation
- Wed. Feb 04, 08:00: **Quiz 1** opens
- Thu. Feb 05, 20:00: **Quiz 1** closes

Week 6 (Feb 08 - Feb 14)
- Tutorial 5: A3 & Bitwise Operators
- Sun. Feb 08, 08:00: **Assignment 3** (Bits and Bytes) Releases
- Sun. Feb 08, 23:59: **Assignment 2** due
- Mon. Feb 09: Lecture 10 Topic: Dynamic Memory De-Allocation
- Wed. Feb 11: Lecture 11 Topic: Dynamic Collections Example

Week 7 (Feb 15 - Feb 21)
- Tutorial: No Tutorials this Week
- Mon. Feb 16: Lecture Cancelled
- Mon. Feb 16: Winter Break
- Tue. Feb 17: Winter Break
- Wed. Feb 18: Winter Break
- Thu. Feb 19: Winter Break
- Fri. Feb 20: Winter Break

Week 8 (Feb 22 - Feb 28)

- Tutorial 6: A4 & Dynamic Memory
- Sun. Feb 22, 08:00: **Assignment 4** (Dynamic Memory) Releases
- Sun. Feb 22, 23:59: **Assignment 3** due
- Mon. Feb 23: Lecture 12 Topic: Building with Makefiles
- Wed. Feb 25: Lecture 13 Topic: Introduction to Processes

Week 9 (Mar 01 - Mar 07)

- Tutorial 7: Dynamic Collections
- Mon. Mar 02: Lecture 14 Topic: Concurrency Overview
- Wed. Mar 04: Lecture 15 Topic: Multithreading

Week 10 (Mar 08 - Mar 14)

- Tutorial 8: Project Planning
- Sun. Mar 08, 08:00: **Final Project** Releases
- Sun. Mar 08, 23:59: **Assignment 4** due
- Mon. Mar 09: Lecture 16 Topic: Multi-Processing and Signals
- Wed. Mar 11: Lecture 17 Topic: Sockets

Week 11 (Mar 15 - Mar 21)

- Tutorial 9: Multithreading
- Mon. Mar 16: Lecture 18 Topic: Multithreading Review
- Wed. Mar 18, 08:00: **Quiz 2** opens
- Thu. Mar 19, 20:00: **Quiz 2** closes

Week 12 (Mar 22 - Mar 28)

- Tutorial 10: Files and I/O
- Mon. Mar 23: Lecture 19 Topic: Scope and Storage Classes
- Wed. Mar 25: Lecture 20 Topic: System Calls, Streams, Files

Week 13 (Mar 29 - Apr 04)

- Tutorial 11: Project Support
- Mon. Mar 30: Lecture 21 Topic: Linking and Libraries
- Wed. Apr 01: Lecture 22 Topic: Memory Review
- Fri. Apr 03: Lecture Cancelled

Week 14 (Apr 05 - Apr 11)

- Tutorial: No Tutorials this Week
- Sun. Apr 05, 23:59: **Final Project** due
- Mon. Apr 06: Lecture 23 Topic: Processes Review
- Wed. Apr 08: Lecture 24 Topic: Extra Topics in C