COMP 2401A (Fall, 2025) Introduction to Systems Programming

Instructor: Yanan Mao (she/her)
Email: yananmao@cunet.carleton.ca
Office Location: HP5270

Coordinator: Vojislav Radonjic (he/him)
Email: vojislavradonjic@cunet.carleton.ca

Teaching Assistants: A list of teaching assistants and their contact information will be posted once the course starts.

Class Information: This course is merged between three sections: **A taught by me**, B and C taught by Connor Hillen.

Lecture Location: Please check Carleton Central for the room location.

Lecture Times: Mondays**,** Wednesdays, 11:35-12:55 (in-person)

Course Website: https://brightspace.carleton.ca

Student hours: The student hours will be posted on the course website beginning Week 2.
Important dates and deadlines can be found here: Registration Dates, including class suspension for fall, winter breaks, and statutory holidays.

Course Information

This is an introduction to system-level programming, including fundamental OS concepts, procedures, primitive data, and 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.

Prerequisite(s): one of COMP 1406, COMP 1006 with a minimum grade of C-.
Postrequisite(s): A minimum grade of C- in COMP 2401 is required to take the postrequisite course. The course list can be found in the Carleton University Calendar.

Learning Material(s) and Other Course/Lab-Related Resources

Textbook: Students are not required to purchase textbooks or other learning materials for this course. We will use COMP2401 Course Notes by Dr. Mark Lanthier, with contributions from Dr. Christine Laurendeau and Dr. Doron Nussbaum.

Laptop Requirement (School of Computer Science) :
Every student who 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. For more information, please visit SCS Laptop Requirement and then review the requirements at Laptop Specifications.

Necessary Software:

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.

1. Downloads & information: https://carleton.ca/scs/tech-support/virtual-machines/

2. Image: Download the appropriate VM image for this course and term when it is made available at the link above.

3. Virtual Box (Free): First, try to use Virtual Box, allowing you to run the virtual machine on your computer [Guide]

4. OpenStack (Free): Optionally (or if you run into issues with VirtualBox), you can try running the virtual machine on the School of Computer Science cloud service. Connecting will require the free Carleton VPN. [Guide]

5. Labs: The virtual machines are also installed in the SCS computer labs.

6. Expectations: You are expected to learn to use the VM on your own. The instructor will not provide technical support. To get operational with the VM, you should take time to read the technical support pages linked above carefully. The first tutorials can help you get started if you encounter issues.

Topics Covered and Learning Outcomes

A common misconception is that this is just a C programming course. C is used to explore lower-level ideas without going 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. **Translates C Language into Executable Code:** Compilation, linking, libraries, system calls

Throughout this course, students will explore foundational topics in system programming, including:

- **To expose students to system-level programming via the C language**
- **To gain experience working with a procedural language**
- **To gain experience working at the low-level implementation of data structures**
- **To gain insight into memory management – Pointers and Memory Addresses**
- **Gain insight into processes and the execution of programs**
- **To gain insight into how programs are structured**
- **To gain experience with Linux/Unix and basic tools**

Assessment Scheme

Grade Breakdown

COMPONENT	GRADE VALUE	DUE DATE	TOPICS
Assignment 1	5%	Sep. 21, 23:50	Chapter 1: C arrays, printing, basic I/O
Assignment 2	5%	Oct. 05, 23:50	Chapter 2, 3: Structs, strings, pointers
Assignment 3	5%	Oct. 26, 23:50	Chapter 2: Bitwise operations
Assignment 4	5%	Nov. 09, 23:50	Chapter 3: Dynamic memory
Final Project	25 %	Nov. 30, 23:50	Chapter 1 - 6
Midterm 1 (Online)	7.5%	Oct. 08, 08:00 - Oct. 09, 20:00	Chapters 1.1 - 3.3
Midterm 2 (Online)	7.5%	Nov 19, 08:00 - Nov 20, 20:00	Chapters 3.3 - 6
Final Exam (In-Person)	30%	TBD by Carleton	Chapters 1 - 7
Engagement Tasks	10% (2.5% bonus)	Various	Practice

Second-Chance Policy

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

Assignments 1 - 4:

1. May be re-submitted and re-assessed within **one week** of when grades are released,
2. Grading times may vary, so the re-submission deadline is not posted in advance and is based on the date that grades are released.
3. The second submission will receive less feedback than the first submission to expedite grading time.
4. The overall mark for each assignment uses the “Weighted Average with Insurance” scheme described below.

Midterm Quizzes 1 and 2:

1. May be re-taken **one week** after the original quiz is released.
2. Class will be cancelled for the first attempt, but will not be cancelled for the second attempt.
3. The overall mark for each quiz uses the “**Weighted Average with Insurance**” scheme described below.
4. Additional requirements to qualify for a resubmission may be added during the term and announced in advance of the first attempt deadline, if they arise.
5. Due to limitations, Brightspace may not accurately reflect the combined mark for an assessment.
6. The final project and final exam are not eligible for resubmission or re-taking and are intended to be the final evaluation of your understanding of the course material.
7. These policies are also the primary short-term accommodation for students who miss the original deadlines for any reason.

Weighted Average with Insurance:

The overall mark for an assessment, A_0, combines the first attempt, A_1, and the second attempt, A_3, as follows:

$$A_0 = max(A_1, 0.85 * max(A_1, A_2) + 0.15 * min(A_1, A_2))$$

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 two results in (0.85 ∗ 95 + 0.15 ∗ 70) = 91.25% for that assessment.

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.

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	GRADE VALUE	MAX	NOTES
Lecture Attendance	2.5%/ half of the 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.

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 marks.

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

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 accommodated only 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.

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., Yanan) 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.

Important Considerations and Expectations:

Land Acknowledgement: Here at Carleton University, it is important that we acknowledge that the land on which we gather is the traditional and unceded territory of the Algonquin Nation.

This course will be delivered **in person**. You are strongly encouraged to attend lectures regularly, as I believe in-person learning is a more engaging and effective way to understand the material. **All relevant course materials**, including lecture slides, solutions, and weekly updates, **will be posted after the lectures.**

Doing well in this course is not only about getting a high grade. It means you really understand the material, and you can connect it with what you have learned before and what you will learn in the future. It is normal to feel uncomfortable when learning something new. But if you can try to challenge yourself a little, instead of always looking for answers directly from AI, you will improve much more. **Real understanding comes from your own thinking and effort.**

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

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.

General expectations for academic integrity in this course include:

1. All work must be completed individually unless explicitly stated in 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:

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 the section about the 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.

Midterm Quizzes 1 and 2; You are permitted to:

1. Work individually and use a non-programmable calculator to assist with calculations.

Final Project; You are permitted to:

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.

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 firsthand 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 on 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 it 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, 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 a real understanding earned by working through the material. Here are a few tips for the use of generative AI, noting that AI in this course is permitted under only a few circumstances described in the Plagiarism Policy Section:

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:
    a. "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.
    b. "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?"

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.

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 Section above 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:** Peer mentoring, workshops, industry events.
- **Carleton Computer Science Society Events:** Workshops, study groups, community events, and more.
- **Centre for Student Academic Support:** Time management, study skills, organization, and general academic skills.

Technical support?

- **Virtual Machine Overview:** Virtual machine overview, VirtualBox Technical Support Guides, OpenStack Technical Support Guides, and SCS Technical Support Procedures.

Less-academic Support?

- **Ombuds Services:** 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.

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: Computer Laboratories. All SCS computer lab and technical support information can be found at: Technic Support. Technical support staff may be contacted in-person or virtually, see this page for details: Contact IT Support.

Mental Health and Wellness

If you are struggling, please do not hesitate to reach out. I am happy to listen, and/or direct you to resources that might help. If you need extra help with course content or happen to miss a class, there's no need to worry—materials will be posted on Brightspace, and I'm happy to arrange additional office hours beyond the scheduled time to help you catch up. Remember that Carleton also offers an array of mental health and well-being resources, which can be found on the Carleton Wellness Website.

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/).

Chat GPT/Generative AI Usage

As our understanding of the uses of AI and its relationship to student work and academic Integrity continue to evolve, students are required to discuss their use of AI in any circumstance not described here with the course instructor to ensure it supports the learning goals for the course.

Many of the assessed activities in this course were designed to be completed by an individual working alone. Unless it is explicitly stated otherwise, the use of any AI system will be considered academic misconduct. This includes, but is not limited to, chatbots or code generators (e.g., ChatGPT, Google Gemini, Microsoft Copilot), research assistants (e.g., Elicit), and image generators (e.g., Stable Diffusion, Dall-E), etc. An exception to the above rule is made for automated grammar and punctuation checking tools (such as Grammarly).

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 Prof. Mao. Please email me or visit during my office 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 the Ombuds services.

Tentative Course Calendar for Section A

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

Week 1 (Aug 31 - Sep 06)

- Tutorial: No Tutorials this Week
- Wed. Sep 03: Lecture 1 Topic: Instructions, Addresses, and Compiled Programs.
- Wed. Sep 03, 08:00: Assignment 1 (Intro to C and Linux) Releases.

Week 2 (Sep 07 - Sep 13)

- Tutorial 1: Welcome to Linux
- Mon. Sep 08: Lecture 2 Topic: Pointers and Addresses
- Wed. Sep 10: Lecture 3 Topic: Numerical Base Conversions.

Week 3 (Sep 14 - Sep 20)

- Tutorial 2: Debugging Pointers and Arrays
- Mon. Sep 15: Lecture 4 Topic: Data Types and Bit Models
- Wed. Sep 17: Lecture 5 Topic: Structs and Unions

Week 4 (Sep 21 - Sep 27)

- Tutorial 3: Bits and Bytes
- Sun. Sep 21, 23:59: Assignment 1 due
- Mon. Sep 22, 08:00: Assignment 2 (Pointers and Arrays) Releases
- Mon. Sep 22: Lecture 6 Topic: Strings
- Wed. Sep 24: Lecture 7 Topic: Bitwise Operators

Week 5 (Sep 28 - Oct 04)

- Tutorial 4: Strings
- Mon. Sep 29: Lecture 8 Topic: Dynamic Memory Allocation
- Wed. Oct 01: Lecture 9 Topic: Dynamic Memory De-Allocation

Week 6 (Oct 05 - Oct 11)

- Tutorial 5: Structures
- Sun. Oct 05, 23:59: Assignment 2 due
- Mon. Oct 06, 08:00: Assignment 3 (Bits and Bytes) Releases
- Mon. Oct 06: Lecture 10 Topic: Dynamic Collections Example
- Wed. Oct 08, 08:00: Quiz 1 opens
- Thu. Oct 09, 20:00: Quiz 1 closes

Week 7 (Oct 12 - Oct 18)

- Tutorial 6: Dynamic Memory
- Mon. Oct 13: Lecture Cancelled (Holiday)
- Wed. Oct 15: Lecture 11 Topic: Building with Makefiles

Week 8 (Oct 19 - Oct 25)

- Tutorial: No Tutorials this Week
- Fall Break, Enjoy!

Week 9 (Oct 26 - Nov 01)

- Tutorial 7: Linked Lists
- Sun. Oct 26, 23:59: Assignment 3 due
- Mon. Oct 27, 08:00: Assignment 4 (Dynamic Memory) Releases
- Mon. Oct 27: Lecture 12 Topic: System Calls, Streams, Files
- Wed. Oct 29: Lecture 13 Topic: Introduction to Processes

Week 10 (Nov 02 - Nov 08)

- Tutorial 8: Files and Streams
- Mon. Nov 03, 08:00: Final Project Releases
- Mon. Nov 03: Lecture 14 Topic: Concurrency Overview
- Wed. Nov 05: Lecture 15 Topic: Multithreading

Week 11 (Nov 09 - Nov 15)

- Tutorial 9: Multithreading
- Sun. Nov 09, 23:59: Assignment 4 due
- Mon. Nov 10: Lecture 16 Topic: Multi-Processing and Signals
- Wed. Nov 12: Lecture 17 Topic: Sockets

Week 12 (Nov 16 - Nov 22)

- Tutorial 10: IPC
- Mon. Nov 17: Lecture 18 Topic: Scope and Storage Classes
- Wed. Nov 19, 08:00: Quiz 2 opens
- Thu. Nov 20, 20:00: Quiz 2 closes

Week 13 (Nov 23 - Nov 29)

- Tutorial 11: Project Support
- Mon. Nov 24: Lecture 19 Topic: Linking and Libraries
- Wed. Nov 26: Lecture 20 Topic: Memory Review

Week 14 (Nov 30 - Dec 06)

- Tutorial: No Tutorials this Week
- Sun. Nov 30, 23:59: Final Project due
- Mon. Dec 01: Lecture 21 Topic: Processes Review
- Wed. Dec 03: Lecture 22 Topic: Extra Topics in C