COMP1005/1405 A (Winter 2026)

Introduction to Computer Science I

Land acknowledgement:

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

Teaching Team:

- **Instructor:** Sean Benjamin (Lecturer, He/Him)
  - **Email:** SeanBenjamin@cunet.carleton.ca
  - **Office:** HP5372
  - **Primary Communication:** Refer to the communication policy below before reaching out.
  - **Call Me:** Sean
- **Lab Coordinator:** Leila Chinaei (She/Her)
  - **Email:** LeilaChinaei@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 This course is merged between **two** sections. Section A taught by Sean Benjamin and Section B taught by Connor Hillen.

- **Lecture Times:** Attendance is expected for all sections.
  - **Section A:** Mon. Wed. 08:35 - 09:55
- **Lecture Locations:** In-person, refer to your schedule.
- **Tutorials:**
  - **Start:** Begin Week 2
  - **Locations:** Refer to your schedule
  - **Times:**
    - **A1**: Mon. 13:05 – 14:25
    - **A2**: Fri. 13:05 – 14:25
    - **A3**: Wed. 14:35 – 15:55
  - **Attendance:** May attend other sections if space permits.
  - **Cancellations:** There will be no tutorials Weeks 1, 7, 13, 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 computer science and programming. Topics include: algorithm design; control structures; variables and types; linear collections; functions; debugging and testing. Special attention is given to procedural programming in a modern language, computational thinking skills, and problem decomposition. Also listed as COMP 1405.

Precludes additional credit for BIT 1400, CGSC 1005, ECOR 1031, ECOR 1041, ECOR 1042, ECOR 1051, ECOR 1606, ITEC 1400, ITEC 1401, SYSC 1005.

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 5.1] Second-Chance Grading Policy
- [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 Textbook (Free, Online):** We will be loosely following the online, interactive textbook **Think Like a Computer Scientist** available for free here: https://runestone.academy/ns/books/published/thinkcspy/index.html
  - Many interactive exercises where you can write Python code directly in the web browser, no account necessary
- **VS Code & Python 3.14:** We will be using the freely available Visual Studio Code as our primary text editor and Python 3.14 as our primary programming language. You will need to setup and install both Python and VS Code.
  - Follow setup Step 1 for your operating system: https://code.visualstudio.com/docs/setup/setup-overview
  - Follow the Quick Start Guide for Python: https://code.visualstudio.com/docs/python/python-quick-start
  - Disable any AI Copilot functionality so that you can build up the muscle memory needed for beginner programmers (See Section 10 for information about AI usage in this course)

3. Note from Instructors

Hello and welcome to COMP1005/COMP1405! 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.

This course is primarily here to teach you about programming and getting started with thinking about data representation, problem solving, and how we implement complex algorithms in programming languages. A fully engaged student will:

- Design and express simple algorithm using flowcharts and pseudocode,
- Implement simple algorithms using the Python 3 programming language,
- Create expressions with arithmetic, logical, and comparative operations,
- Create branching and repeating control structures, with and without nesting,
- Explain variable assignment, primitive data types, and the basics of computer memory,
- Design and implement functions and explain function scope and recursion,
- Create, access, and manipulate linear, multidimensional, and associative collections,
- Implement and discuss the efficiency of some basic sorting and searching algorithms.

I believe that all students can reach these goals. This content can be challenging for many folks and if you feel you are alone in falling behind, please know that many different people are facing challenges and you are encouraged to reach out for support during student hours for help getting back on track. Even for experienced professionals, a lot of programming involves confusing and frustrating errors, a need to go back to planning, and unexpected gaps in knowledge.
Ask questions in class, attend student hours, and try to use each error as a learning opportunity.

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
Assignment 1	5%	Jan. 28, 17:00
Assignment 2	5%	Feb. 25, 17:00
Assignment 3	5%	Mar. 11, 17:00
Assignment 4	5%	Mar. 25, 17:00
Quiz 1	7.5%	Feb. 02, in-class
Quiz 2	7.5%	Mar. 02, in-class
Quiz 3	7.5%	Mar. 16, in-class
Quiz 4	7.5%	Mar. 30, in-class
Final Exam	40%	Scheduled by Registrar
Engagement Tasks	10% (2.5% bonus)	Various; Last day to submit to forum: Apr. 02, 23:59

5.1. Second-Chance Policy

To promote learning through mistakes and feedback, this course will be permitting the re-submission of Assignments 1 - 3 after feedback is released so that you can learn from any mistakes, revise, and possibly improve your mark:

- **Assignments 1 - 3:**
  - 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.
  - **NOTE:** Assignment 4 can **NOT** receive a second-chance to have sufficient time for grading.
- Due to limitations, Brightspace may not accurately reflect the combined mark for an assessment
- **These policies are the short-term accommodation for students who miss the original deadlines for any reason.**

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:

$$A_o = \max(A_1, 0.80 \times \max(A_1, A_2) + 0.20 \times \min(A_1, A_2))$$

- 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 80% of the highest mark of the two attempts and 20% of the lowest of the two attempts.
- Thus, if you do not take the first attempt, your maximum score is 80% assuming a perfect score on the second attempt.
- E.g., A score of 70% on attempt 1 and 95% on attempt 2 results in $(0.80 * 95 + 0.20 * 70) = 90$% for that assessment

5.2. Re-weighting Policy

- **Quizzes:** A higher score on the Final Exam will automatically replace any lower Quiz scores.
- **Note:** This re-weighting, while automatic, should only be a **last resort** under extenuating circumstances. Moving weight to the final exam both reduces your understanding of the material, reduces opportunities for feedback, and can makes the final exam worth a very large amount.

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	3% / half of semester	6%	Based on participation in in-class Wooclap polls. Up to 6% available, once before the break and once after. 3% is awarded for responding to at least 80% of Wooclap polls before the break and 3% is awarded for responding to at least 80% of Wooclap polls after the break. No marks for attending fewer than 80%.
Tutorial Attendance	0.5% / lab	5%	Based on submission of a password to Brightspace quizzes. Password is given by the TA during each tutorial.
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, 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

You do **not** need to request any accommodation for these assessments as the existing policies already account for missed work.

- **Assignments 1-3:** You may submit to the second-chance submission following the release of grades for the assignment.
- **Assignment 4:** Short-term accommodations or long-term accommodations may be available by emailing the instructor for valid extenuating circumstances such as medical incapacitation.
- **Quizzes:** A higher mark on the final exam can replace the lower marks of any quizzes.
- **Engagement Tasks:** There are more engagement tasks available than marks required, so you can do alternative tasks.

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 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?**
  - 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. Do **not** expect responses on weekends or after 5:00PM (17:00).

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.

- Keep the titles of your posts clearly informative to what you are asking so that other students can find responses easier
- Posts that ask the same question as another post will be remove to keep the forum clearly organized.
- **Note:** Discussion forums are **NOT** meant to be an immediate help desk for any minor issues that you encounter while working. Please take the time to review the specification again and take some time to learn how to plan and debug.

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:** There are no reasons to directly email a teaching assistant. You may discuss with TAs during tutorials, posting to the grade dispute forum, or posting to the Brightspace discussion forum as appropriate.
- **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., Obiwan, Mr. Kenobi 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. Emails which are answered in-class, via announcement, or ask about basic info from the course outline or specification (e.g., dates and deadlines which are posted) may not receive a response.

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 **two business days** 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. Technical issues, 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 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 for you.

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.** This includes last-minute Brightspace issues. The computer labs are available to work in if you experience technical issues with your computer and you should give yourself enough time to use this 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.
3. **Backup and Submit Work Often:** It is very easy to accidentally delete or corrupt your files. One way to maintain backups of your work is to submit often to Brightspace as well as using 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/

   Frequent submissions allow us to grade a previous submission if you accidentally break the latest submission at the deadline. The lab coordinator **must** be informed via email that you would like the earlier submission assessed within 24 hours of the deadline.
4. **Assignments Need Functioning Code:** Code which does not run fully and correctly 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 runs 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 assignments** 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., additional text 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**; 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. While AI usage risks reducing your learning, as described in Section 10, if you choose to use AI, make sure that you limit AI usage to a few lines of code here-and-there and cite any usage, avoiding having an AI write entire pieces of functionality. Programming without AI assistance **is** a form of studying for exams and retaining the course material.

2. **Assignments 1 - 4, Homework, Tutorial Exercises**; Requirements and Restrictions:
   1. In any case where you receive support from another source, you **must** cite that source in the comments of the code.
      - Describe the source of your assistance and the capacity that they assisted you, excluding TAs, provided course materials, and the teaching assistants
   2. 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 able to explain your work when asked.
   3. You should never take code from another student and include it with your submission.

3. **Quizzes 1-4**; You are permitted to:
   1. Work individually and use a non-programmable calculator to assist with calculations.

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. 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 run my Python 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/.

School of Computer Science Laptop Requirement: Every student that has been enrolled in a 1000-level (i.e., first year) course offered 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/.

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 (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-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: Course Introduction, CS, Computers
- Wed. Jan 07: Lecture 2 Topic: Problem Solving

Week 2 (Jan 11 - Jan 17)

- Tutorial 1
- Mon. Jan 12: Lecture 3 Topic: Algorithms
- Wed. Jan 14: Lecture 4 Topic: Python, Variables, Data Types
- Thu. Jan 15, 08:00: **Assignment 1** (Basics) Releases

Week 3 (Jan 18 - Jan 24)

- Tutorial 2
- Mon. Jan 19: Lecture 5 Topic: I/O, Operators
- Wed. Jan 21: Lecture 6 Topic: Logic and Conditions

Week 4 (Jan 25 - Jan 31)

- Tutorial 3
- Mon. Jan 26: Lecture 7 Topic: Debugging and Testing
- Wed. Jan 28: Lecture 8 Topic: Functions and Scope
- Wed. Jan 28, 17:00: **Assignment 1** due
- Thu. Jan 29, 08:00: **Assignment 2** Releases

Week 5 (Feb 01 - Feb 07)

- Tutorial 4
- Mon. Feb 02, 08:00: **Quiz 1** in-class
- Wed. Feb 04: Lecture 9 Topic: Looping

Week 6 (Feb 08 - Feb 14)

- Tutorial 5
- Mon. Feb 09: Lecture 10 Topic: Lists
- Wed. Feb 11: Lecture 11 Topic: Strings

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
- Mon. Feb 23: Lecture 12 Topic: Files and Exceptions
- Wed. Feb 25: Lecture 13 Topic: Multi-Dimensional Lists
- Wed. Feb 25, 17:00: **Assignment 2** due
- Thu. Feb 26, 08:00: **Assignment 3** Releases

Week 9 (Mar 01 - Mar 07)

- Tutorial 7
- Mon. Mar 02, 08:00: **Quiz 2** in-class
- Wed. Mar 04: Lecture 14 Topic: Dictionaries

Week 10 (Mar 08 - Mar 14)

- Tutorial 8
- Mon. Mar 09: Lecture 15 Topic: Searching
- Wed. Mar 11: Lecture 16 Topic: Sorting
- Wed. Mar 11, 17:00: **Assignment 3** due
- Thu. Mar 12, 08:00: **Assignment 4** Releases

Week 11 (Mar 15 - Mar 21)

- Tutorial 9
- Mon. Mar 16, 08:00: **Quiz 3** in-class
- Wed. Mar 18: Lecture 17 Topic: Recursion

Week 12 (Mar 22 - Mar 28)

- Tutorial 10
- Mon. Mar 23: Lecture 18 Topic: Recursion
- Wed. Mar 25: Lecture 19 Topic: Object-Oriented Programming
- Wed. Mar 25, 17:00: **Assignment 4** due

Week 13 (Mar 29 - Apr 04)

- Tutorial: No Tutorials this Week
- Mon. Mar 30, 08:00: **Quiz 4** in-class
- Wed. Apr 01: Lecture 20 Topic: Java and Other Languages
- Fri. Apr 03: Lecture Cancelled

Week 14 (Apr 05 - Apr 11)

- Tutorial: No Tutorials this Week
- Mon. Apr 06: Lecture 21 Topic: TBA
- Wed. Apr 08: Lecture Cancelled