Course Syllabus

Welcome to CSC148! This course, *Introduction to Computer Science*, introduces you to how computer scientists think in a systematic way about computing. Our hope is to provide you the basics for approaching program design principles such as encapsulation, modularity, and information-hiding, comparing different program implementations for efficiency, and building powerful data structures. Please visit the course web page and discussion board often, and read email sent to your University of Toronto email for important announcements.

For all course-related questions, please contact csc148h5-2025-instructors@cs.toronto.edu

Learning Outcomes

By the end of this course students will be able to:

- explain and operate with the memory model in Python when designing programs
- operate with the basics of program design principles such as encapsulation, modularity and information-hiding
- building powerful data structures and using them for specific goals (e.g., stacks, queues, linked lists, trees, binary search trees, expression trees, etc.)
- operate with the recursion technique and recognize its applicability to specific programs
- assess computational complexity and compare the efficiency of various programs or algorithms
- assess the correctness of a program using tracing and debugging skills
- design tests to assess that a program meets the intended specifications

Logistics

The course instructors are:

- Prof. Bogdan Simion
  Email: bogdan@cs.toronto.edu
  Office: DH-3076

- Prof. Rutwa Engineer
  Email: rutwa.engineer@utoronto.ca
  Office: DH-3013

- Prof. Marc De Benedetti
  Email: marc.debenedetti@utoronto.ca
  Office: DH-3032

- Prof. Romina Piunno
  Email: romina.piunno@mail.utoronto.ca
  Office: DH-3019

Lectures (rooms on Acorn)
LEC0101: MO+WE+FR 9-10 **LEC0102**: MO+WE+FR 10-11 **LEC0103**: MO+WE+FR 13-14 **LEC0104**: MO+WE+FR 14-15 **LEC0105**: MO+WE+FR 16-17 **LEC0106**: MO+WE+FR 17-18
Labs
PRA0101: (TH 9-11) **PRA0102**: (TH 9-11) **PRA0104**: (TH 11-13) **PRA0105**: (TH 11-13) **PRA0106**: (TH 11-13) **PRA0107**: (TH 13-15) **PRA0108**: (TH 13-15) **PRA0109**: (TH 13-15) **PRA0110**: (TH 15-17) **PRA0111**: (TH 15-17) **PRA0112**: (TH 15-17) **PRA0113**: (TH 17-19) **PRA0115**: (TH 17-19) **PRA0116**: (TH 19-21) **PRA0118**: (FR 9-11) **PRA0121**: (FR 11-13) **PRA0122**: (FR 11-13) **PRA0123**: (FR 11-13) **PRA0124**: (FR 13-15) **PRA0125**: (FR 13-15) **PRA0126**: (FR 15-17) **PRA0127**: (FR 15-17)

Notes:

- Our first lecture is on Monday, January 6.
- You MUST go to the lecture section you are enrolled in, otherwise we cannot guarantee we can accommodate you and you may be asked to leave. We have limited seating so if the lecture room gets severely crowded it can affect our ability to help with problem-solving the students who are enrolled in it.
- Labs start in the first week of classes on Thursday, January 9 and Friday, January 10 (depending on which PRA you are enrolled in).
- Lab room assignments are according to your ACORN enrollment. You MUST enroll in a PRA

immediately to guarantee that you can get a spot in a PRA that does not overlap with your LEC section!

- You MUST attend the PRA that you are enrolled in, to receive a grade for your lab work. You MUST NOT go to a different PRA section, as you will likely not get a seat, you will not be on the TA's list of students, and we will not have a quiz ready for you! It is implied that if you do not enroll in *any* PRA section, you will get a 0 on the labs.
- All lectures and labs begin at ten minutes past the hour.

Official Communication:

- The only sources of official information are the course instructors. For some matters, only the course coordinator may advise you or provide clarification. Unofficial communication from classmates or TAs (for instance, Discord, WeChat, Slack, Teams, email, etc.) is not adequate advice.
- On Piazza, answers from classmates may not be correct. Wait for an instructor or designated help TA to confirm. It takes time for us to answer, there are close to 1000 students enrolled!
- Lab and Lecture TAs are not supposed to answer assignment or test related questions. This is regardless if they are tasked with grading such items! Focus on lecture work and lab work during the LECs and PRAs, respectively!
- TAs might not have accurate answers on course logistics. When in doubt, ask instructors on Piazza or come to office hours!

Office hours

Instructor office hours: Instructor office hours will **in-person**. Please make sure to show up during the indicated times below at the instructor's office throughout the term:

- Rutwa: Tuesdays 10am-1pm (in person)
- Bogdan: Mondays 2:30-4pm, Fridays 2:30-4pm (in person)
- Marc: Wednesdays 10:30am-12:30pm, 3-4pm (in person)
- Romina: Mondays 3:30-5pm, Fridays 3:30-5pm (in person)

Instructor office hours will begin **after the first week of classes**. If you have questions during the first week, please email the instructors, or talk to us after lecture.

Instructor office hours *may* shift online, if necessary. This may change throughout the term and you will be notified, depending on unexpected circumstances, in order to hold office hours in a safe manner.

Help Centre

To provide more support for your learning, we may also providing additional office hours in the form of a Help Centre, which will be staffed by course TAs, who can help you with any questions on course content related to large assessments, such as assignments.

The Help Centre will only run for part of the timeframe for assignments 1 and 2. We will likely schedule the Help Centre to avoid overlap with all LECs and PRAs. Tentatively, this will likely be on

MO+TU+WE 11-13, and FR17-19, but further details and exact timeslots will be announced later in the term.

IMPORTANT NOTE: The TAs' role is to help you with assignment clarifications where you might find something unclear or to assist you with problem solving. They **will not solve the problems for you**, they will merely guide you to figuring out the problems on your own. For help with code problems or bugs, you must explain your work AND what you've done to debug your code and where you got stuck during debugging. **If you have not even attempted to debug your code, the TA will not debug your code for you.** While the TA may help you with how to use a debugger in general, you must demonstrate that you've done your due dilligence, since knowing how to debug your own code is an important part of the learning process.

During busy times, to help as many students as possible, each turn will be limited to 5-10 minutes max. If you need further help, you will have to get back in the queue, so that we give a fair chance to everyone to get some assistance from the Help Centre TAs. Please note that during very busy times, if the help queue is very long, you may not get your turn, so it is advisable that you do not wait until the last few days before the deadline to ask questions.

Creating a positive learning environment

We are committed to creating a respectful learning environment in computer science courses for all students and expect that you will adhere to the University of Toronto Code of Student Conduct. Please be mindful of how your behaviour influences the atmosphere in our learning community, not just in classes, but also in computer labs, in online forums, and anywhere that you interact with other students and members of the department.

Prerequisites

- CSC108. We'll be using Python3.11 in the course, but comfort with other imperative programming languages like Java or C is an added bonus. We strongly recommend you revise the CSC108 materials, as we will expect that the content from CSC108 is prior knowledge that you must have.

Textbook

We'll be updating a set of lecture notes throughout the term; this is your required reading for this course.

We might occasionally post links to supplementary readings from this online free textbook: http://openbookproject.net/thinkcs/python/english3e.

Software and labs

For information about the software we'll use for this course, and the departmental labs in Deerfield Hall (which are available to all CSC148 students, should we be back in person), please see the Software page.

Assessment

	9 preps
	Sundays at 5pm, hard deadline: Mondays at 9am
	8%
	10 labs
	At the end of your lab section
	4%
	Assignment 1
	February 17 at 5pm
	9%
	Midterm
	February 24 at 7-9pm
	20%
	Assignment 2
	March 31 at 5pm
	9%
	Final Exam
	April exam period
	50%

Preps are all mandatory to make sure you are prepared to work on problem-solving in lecture. Prep 1 is not for credit, while the rest of the 9 preps are graded. We will take your **best 8 out of the 9 graded preps for 1% each, totalling 8%**.

A **bonus of 1%** may be provided for filling in two surveys (one at the start of term for 0.5% **and** one at the end of term for 0.5%). These surveys refer to a study we are running, which we will be providing more information on in the course. As these surveys are optional, this bonus is not part of the 100%, but **can make up for lost marks in the weekly preps**. By completing these surveys you will help us improve your learning experience and that of future students in computer science.

Labs are important to further develop skills so you must participate in the labs each week. We will take the **best 8 out of 10 labs for marks, for 0.5% each, totalling 4%**.

Notes:

- **You need to score at least a 40% on the final exam to pass the course.** Students scoring less than 40% on the final exam will receive a maximum final grade of 47%.

Assignment Policies

Assignments must be submitted electronically, using the MarkUs online system. Be sure to confirm that you have submitted all the required files and the correct version of each; we cannot accept missing files or a different version of an already-submitted file after the due date.

Code that you submit to us **must work on the lab machines or in a similar environment that we provided for setting up your own computer (see the Software page)**, in order to earn credit, as we will be testing your code on a similar environment.

Assignments are **individual work** and are due at **5:00 p.m.** on the deadline date, as we want course staff and tech support to be available at the deadline in case of issues.

One of the assignments has an associated writing component. You will receive feedback on the writing component and will have an opportunity to address feedback and resubmit to improve your grade

Individual Work Policy

All assignments and weekly preps must be done individually.

Please note the academic integrity policy and do not commit academic offences, as we take such cases seriously and the penalties are severe.

Please note the policy on use of generative AI as well.

Late Assignments or Preps

MarkUs is known to be slow when many students try to submit right before a deadline. **Aim to submit your work at least one hour before the deadline.** You can submit your work more than once—only the last version submitted before the deadline will be graded. Do not leave things last minute! For special consideration for assignments, please review the Special Consideration section.

For weekly preps, **no late submissions past the hard deadline will be graded**, under any circumstances. The whole point of the preps is to be ready for active problem-solving in lecture. Additionally, we often may discuss prep content in class (potentially including partial solutions) and it would be unfair if students in earlier sections in the day/week are at a disadvantage.

Lab Policies

For CSC148, labs are scheduled time for you to get hands-on experience applying the concepts you learn each week. Each one consists of a few tasks that range from design on pencil-and-paper, implementing functions in Python, writing tests, and even learning some new material! There will also be an informal quiz activity at the end of lab, to give you some experience working in an exam-like environment as the quiz will test your individual grasp of the material.

Please read through the lab logistics and participation guidelines carefully so that you are aware in advance of the expectations.

Lab logistics:

Please consult ACORN to find your assigned lab section and room. The list of lab sections and rooms is also provided under the Logistics section. If you are not enrolled in a lab (PRA) section yet, please make sure to do so asap, otherwise you will get a 0 on the labs.

To avoid situations where some students do not get a seat in the lab, you **must attend the lab room and timeslot that you enrolled in on ACORN**.

NOTE: **If you show up to a PRA you are not enrolled in, your work will NOT be graded, as you will not be on the TA's grading list for that lab section, and we will not have a quiz printed for you.**

Lab participation and other guidelines:

You are not marked on the work you complete. Instead, labs are marked **participation in three parts**: the programming activity (which is done with partners), the individual quiz, and the discussions on the quiz. As long as you do the following, you will get the lab grade each week:

- Join the lab within at most **10 minutes** of it starting (20 minutes past the hour is the "cutoff")
- Put in a solid effort each lab (not just goofing off) during the lab activity, ask questions if you need assistance from the TA. Consider this as valuable practice time that we provide you, with live assistance readily available.
- Complete the quiz activity at the end of the lab
- Participate in the discussions, to filter out any misconceptions. It's best if you do this NOW, to make the most out of the lab and not fall behind.

NOTE: **Labs are not for asking assignment questions, they are dedicated for working on the lab activity!**

Pair Programming

In your labs, you'll usually work in pairs. This gives you the opportunity to practice *communicating* in computer science, which most new students do not have experience with and which is very important to develop. By working in pairs, you'll get to:

- meet new people to form study groups later on
- ask a question without waiting for your TA to respond

- share your thoughts with a peer rather than a teacher (no pressure!)

During programming tasks, only one student in each pair should be doing coding at any one time, but of course **both** students **must** be actively engaged. Here's a brief description of the two roles in an effective pair programming team:

- **Driver**: the one who is touching the keyboard. Responsible for most minor implementation details. *Must verbally explain each line of code written, as they write.*
- **Navigator**: the one who is watching for mistakes, and thinking longer term. *Must stop whenever the driver writes or says something unclear.*

Drivers and navigators should switch frequently: about once every 10 or 15 minutes. As a team, both partners have a responsibility to make sure the other is learning. This might slow you down, but that's okay. *You earn your participation mark as long as the TA can tell you've made an honest effort*.

Lab Tips

1. Your TAs are there to help! Don't be afraid to ask them questions about the lab - that's why they're there.

2. If you happen to finish the regular lab tasks early, you have a few options:

   - Discuss the lab exercises with your partner, to make sure you both of you are not left with misconceptions.
   - Try to implement a different approach, where applicable. Thinking of alternative solutions is good practice.
   - Help the other students, where applicable! This is a great way to get to know other students and make sure you really understand the material.
   - Work on the additional and/or challenge exercises on the lab.

3. We tend to give *more* work than can be completed in a two-hour period, so that you have lots of opportunities for practice even after the lab is over. Don't feel bad if you don't complete a lab, try to finish each one for homework!

4. While you may be able to access the lab handouts ahead of your lab's start time, there's really no benefit to doing so. Remember that the primary goal of the lab is for you to learn with your TA there, and work with another student, and that you earn marks based on participation, not completion.

5. Participation grades will appear on MarkUs around a week (or a few weeks, depending on busy times during the term) after the lab has completed. **Please wait for at least 1-2 weeks before asking about a missing lab participation grade.**

Special Consideration

For **preps**, no late submissions will be accepted, as some flexibility is built into the marking scheme as well as the grace period between between Sunday 5pm and the hard deadline of Monday 9am when the first LEC starts.

For **labs**, they are intended for participation in person in the lab activities, lab quiz and discussions, so no accommodations are possible to get graded for these outside the lab. If you have to miss a lab or two, some flexibility is built into the marking scheme already to account for special situations like illness or severe personal circumstances.

For **assignments or tests**, if you are unable to complete an assignment or if you miss a test due to major illness or other circumstances completely outside of your control, special consideration may be possible. Get in touch with us immediately if you want to receive special consideration.

In order to receive special consideration, you must fill out a Request for Special Consideration Form. Email the completed form to **the instructor mailing list** right away, together with your supporting documentation and absence declaration from Acorn in pdf format.

IMPORTANT: You must get in touch with us **ahead of the assignment due date, or within 24 hours after the deadline at the latest**. Similarly, for a missed midterm you need to inform us **ahead of the test, or within 24 hours after the test at the latest**. You can contact us even before you have filled the special consideration form; we won’t be able to tell you at that point what accommodation you may receive, if any, but can answer other questions and offer advice. It is always easier to resolve situations earlier rather than later.

For the **midterm**, we may not be able to schedule a make-up midterm, given that every student’s schedule is different and we cannot schedule a substantial number of distinct midterms scheduled on different days and times. As such, the only fair option would be to shift the weight of the midterm to the final exam, which is cummulative anyway.

Remark Requests

If you feel there was an error in the marking of an assignment, you may request a remark directly on MarkUs.

If you feel there was an error in the marking of your midterm, you may request that it be remarked via a remark request form. Fill out and submit a Remark Request Form.

You must give a specific reason for the request, referring to a possible error or omission by the marker. Stating the specific potential grading errors for your remark request is **mandatory for us to consider your request**. However, we will **review your entire work, not just the items you pointed out**.

Please keep in mind that your grade **may stay the same, may increase, or may even decrease**, after your remark request is assessed.

Please note that we apply the same grading scheme uniformly for everyone, and that assignments are auto-graded using comprehensive tests for correctness, coding style, etc. Therefore, for fairness to everyone, we cannot give you more marks simply because you disagree with the marking scheme, unless a test happens to be incorrect according to the assignment specs (which we will do our best to rigurously ensure it does not happen).

For prompt turnaround, remark requests **must be received within one week** of when you received the grade for that item.

Academic Integrity

The work you submit must be your own. It is an **academic offence to copy someone else's work, with or without changes.** This includes their code, their words, and even their ideas. **Whether you copy or let someone else copy, it is an academic offence.** The department uses software that compares programs for evidence of similar code. **Academic offences are taken very seriously.**

At the same time, we want you to benefit from working with other students. Obviously, work done with your partner on the labs is a joint effort. You are also welcome to work *appropriately* with other students. To clarify, it is appropriate to discuss course material and technology related to assignments, and we encourage you to do so. For example, you may work through examples that help you understand course material or a new technology, or help each other configure your system to run a supporting piece of software. You may also discuss assignment requirements.

However, **collaboration on preps or assignment solutions is strictly forbidden**. The most certain way to protect yourself is not to discuss solutions or the ideas behind them with other students. Certainly you must not let others see your solutions, even in draft form. Please don't cheat. We want you to succeed and are here to help if you are having difficulty.

Once again, all of the work you submit must be done only by you (and your partner, in case of labs). Plagiarism is academic fraud and is taken very seriously. Please read the Rules and Regulations from the U of T Code of Behaviour on Academic Matters.

Important: Please note that while we don't prohibit the use of private tutors, you have to be aware of what is good tutoring and bad tutoring and stay away from tutors who will do you a disservice and hurt your academic standing.

1. If you have a tutor, they are not allowed to help you solve the assignments or preps! That falls under **unauthorized help** under the Code of Behaviour on Academic Matters.
2. Tutors may show you how to debug code in general, or how to read errors in general and interpret them. However, they are **not allowed to tell you how to solve a problem or what code to write**.
3. If your tutor is telling you what to write or how to solve a problem instead of providing guidance to solving the problem yourself, then they are not helping you. Students who receive such tutoring are likely to fail the course, as they are not learning and will likely fail the midterm and final exam.
4. Most tutors help others students as well, so if your tutor is telling you what to write in your solution, it is possible your assignments will be flagged for plagiarism if other students write similar code, due to the same tutor's bad practices.

Use of Generative AI

In general, students may use artificial intelligence tools for learning course material, practicing with concepts, data structures, or algorithms, getting support to debug or test some piece of code (without getting direct code snippets generated for you!), etc.

However, **all submitted work (i.e., any graded assessments) must be the original work**

produced by the individual student alone.

For graded assessments, you **may** use Generative AI tools **only** if it helps to speed up your work for boilerplate code (like reading from a file or writing to a file), but **not** for tasks that are part of the assignment learning objectives! **You must cite the use of such tools for such purposes, including:**

a. what tool you used, and
b. what prompts you used, and
c. whether the result was correct and complete and if it wasn't, why not, and
d. follow-up prompts if applicable, and
e. briefly what you had to do to alter the code that the tool produced.

Submitting material produced by GenAI tools **without appropriate citation and steps a)-e) above, will be considered plagiarism**. Submitting work with hallucinations (as Large Language Models are known to do) will be considered fabrication. Both are academic offenses.

Please note that midterm and the exam in this course will not allow the use of generative AI tools as aids, so you must be prepared to solve conceptual or coding problems, without assistance. As a result, it is highly important that generative AI be used only as a learning aid and that **you** can produce solutions to problems **on your own**.

If you are not certain or have any question about the use of artificial intelligence applications or tools in this course, please speak with the course coordinator.

Mathematical & Computational Sciences
UNIVERSITY OF TORONTO
MISSISSAUGA

For general course-related questions, please use the discussion board.
For individual questions, accommodations, etc., please contact
the **csc148h5-2025-instructors at cs.toronto.edu** email.
Make sure to include CSC148 in the subject, and to
state your name and UtorID in the email body.