

Course Syllabus



Hello!

Welcome to CSC110! This course, and its follow-up course CSC111, serve as a first-year introduction to the rich and wonderful field of computer science. In this course, you'll learn key programming and theoretical foundations of computer science, and get a taste of how they can be applied to many different areas of computer science, and society at large. We hope you have a great time in CSC110 and CSC111 this year, and are very much looking forward to teaching you!

The material posted on Quercus is required reading. It contains important information: assignment handouts, the policy on missed work, links to all course tools, the announcements page, and more. You are responsible for all announcements made in lecture and on Quercus.

Please read through this course syllabus carefully to familiarize yourself with the content, logistics, and policies of CSC110.

All course announcements will be made on Quercus on the [Announcements](#) page. You are responsible for reading all course announcements.

How to get in touch with us

What not to do

Do **NOT** use Quercus messaging for anything related to CSC110. Your message will likely not be received. Do **not** send emails to course instructor's personal email addresses either, where they are likely to get lost in our usually flooded inboxes. Instead you can do any of the below, depending on your situation –

For personal issues/emergencies

To contact the course instructors regarding personal issues and emergencies please use this email address: csc110-2023-09@cs.toronto.edu (<mailto:csc110-2023-09@cs.toronto.edu>) This is only for *personal questions* (making appointments, remarking requests, missing class, etc.).

Sign your email with your **full name, student number, and UTORid**.

We will try to respond to email by the end of the next business day. However, it may take longer, especially near due dates. If you do not hear back after a few days, please do not hesitate to send a

follow-up email.

Tip: sometimes students may be nervous about emailing a professor for the first time. We've prepared a [an advice page on emailing your professors](#) that you might want to check out!

General course-related questions

For general course-related questions such as clarifying a concept, asking about an assignment, etc., please use Ed Discussion (link on the Quercus sidebar).

We'll be using Ed Discussion as our course online discussion forum and chatroom. Please post all of your questions about the course material and assignments on Ed so that everyone can benefit from your questions. We will monitor the discussion board regularly, but please answer questions from other students—helping someone else learn is one of the most effective ways of truly mastering a subject.

General questions, personal questions, and sometimes just to chat

Last but not last, come see us during our weekly office hours – more on that [here!](#) Regular schedule is in the table below; you can come in on a drop-in basis with any general course-related questions (or – during less busy weeks – even just to chat).

Instructor	Sadia Sharmin (Course Coordinator)	Tom Fairgrieve
Office Hours	Tue 2:10pm - 3:00pm (in-person) Thur 2:10pm - 3:00pm (in-person) Weds 4:10pm - 6:00pm (online*)	Mon 2:10pm - 4:00pm (in-person) Tue 1:10pm - 2:00pm (in-person) Thur 1:10pm - 2:00pm (in-person)
<p>Note: all in-person office hours will be held in BA 4290 (https://map.utoronto.ca/?id=1809#!m/494470). *Online office hours will be held on Zoom at https://utoronto.zoom.us/j/84245010690 (https://utoronto.zoom.us/j/84245010690) (Passcode: 110110)</p>		

What is on this page

There is quite a lot of information on this page, and we get it, reading this much at once can be hard! So, we are splitting it up for you in multiple sections, with links to each section below:

- [Lectures](#)
- [Office Hours](#)
- [Tutorials](#)
- [Assessments](#)

- [Preps](#)
- [Assignments](#)
- [Tests](#)
- [Technology requirements](#)
 - [Software setup](#)
 - [The Department of Computer Science Teaching Labs](#)
- [Textbooks and resources](#)
- [Accommodations and accessibility services](#)
- [Special consideration for term tests](#)
- [Special consideration for other homework](#)
- [Remark requests](#)
- [CSC110 Community Code of Conduct](#)
- [Copyright notice](#)
- [Switching to CSC108](#)

We also highlighted key parts throughout. :)

What is this course about?

An introduction to the field of computer science combining the tools and techniques of programming (using the Python programming language) with rigorous mathematical analysis and reasoning. Topics include: data representations; program control flow (conditionals, loops, exceptions, functions); mathematical logic and formal proof; algorithms and running time analysis; software engineering principles (formal specification and design, testing and verification). Prior programming experience is not required to succeed in this course.

This course is restricted to students in the first year Computer Science admission category, and is only offered in the Fall term. Other students planning to pursue studies in computer science should enrol in CSC108H1, CSC148H1, and CSC165H1/CSC240H1.

Learning outcomes

In this course, you learn to:

1. Analyze a problem domain written in English; represent key definitions and properties using mathematical logic; and design, implement, and evaluate computational solutions to solve a problem.
2. Understand and write programs using standard features of the Python programming language.
3. Understand and use a variety of professional software development skills, including: programming using an Integrated Development Environment (IDE); writing clear documentation; debugging and

testing programs; reading technical documentation and source code to learn how to use an external program or library.

4. Analyse the running time of a program.
5. Define and implement common abstract data types and algorithms.
6. Create a mathematical proof or disproof of a given statement in new and familiar domains, choosing from among different proof techniques to use. Apply proofs of mathematical statements to justify the correctness of algorithms.

Lectures

The first lecture is on **Thursday, September 7**. All lectures start at "U of T time", which is 10 minutes past the hour, and end on the hour. This allows for 10 minutes of travel/break time if you have back-to-back classes.

Lectures	LEC0101	LEC0201
Meeting Time and Location	Mondays 11:10am - 1:00pm (Location: MY 150 (https://map.utoronto.ca/?id=1809#!m/494495 .)	Mondays 11:10am - 1:00pm (Location: MY 150 (https://map.utoronto.ca/?id=1809#!m/494495 .)
	Tuesdays & Thursdays 9:10am - 11:00am (Location: WB 116 (https://map.utoronto.ca/?id=1809#!m/494460 .)	Tuesdays & Thursdays 3:10pm - 5:00pm (Location: HS 610 (https://map.utoronto.ca/?id=1809#!m/494459?s/ .)
Instructor	Tom Fairgrieve	Sadia Sharmin (course coordinator)

Note: On Mondays, both lecture sections will be combined into a single room, MY 150. This will let you get to meet and interact with students across lecture sections, and allow us to run events for the entire first-year computer science cohort.

Our lectures will alternate between instructors presenting new concepts and you actively engaging with course material through problem-solving exercises. Attendance in lecture is not graded, but is considered a mandatory part of the course.

Lecture recordings

LEC0101 will be participating in the University of Toronto's *OpenCast Content Capture Pilot*, which will automatically record lectures and post them on the [OCCS Student App](#). However, because of the amount of active learning that will take place during lecture, please note that simply watching these videos is *not* a substitute for attending class! Our recommendation is to use these recordings for review

purposes only, or if you miss a lecture due to extenuating circumstances. If you did miss the lecture, we strongly recommend working through the in-class exercises (which are posted separately on Quercus) when we reach those points in the lecture, so that your experience is as close to the live classroom experience as possible. These recordings are meant for your personal learning, and you may *not* distribute these recordings or make your own (please see the [Copyright notice](#) below).

Office hours

Each week, your instructors will hold drop-in office hours that provide an informal setting for students to drop in and ask questions or just chat about the course material (or, during less busy weeks, we can just chat about cats, board games, and other good things in life). You are welcome to attend any of the scheduled office hours (Please attend, even just to say hi! We don't bite, and we get lonely during the first few weeks when everyone seems to be scared of approaching professors...)

Our office hours are *student-driven*, meaning teaching team members won't have any material prepared. Instead, the discussion will be based on whatever questions you'd like to ask. Most office hours are also *group-based*, meaning we generally stick to questions that aren't specific to any particular student, but rather to course concepts and answers that every student can benefit from.

If you have a personal matter to discuss, you may attend Sadia's online office hour session on Wednesday where you can request a private breakout room for a one-on-one conversation. If you have such a matter to discuss but the online times do not work with your schedule, or you wish to speak with Tom, feel free to send an email to the course account to request an appointment (let us know your availability for the week within the email, being as flexible as possible).

Tutorials

At the end of each week, you will participate in a two-hour tutorial, which is an opportunity to reinforce and extend your learning from lecture that week.

NO TUTORIALS DURING THE FIRST WEEK OF CLASS. The first tutorial is **Friday, September 15**. Like lectures, all tutorials start at "U of T time", which is 10 minutes past the hour.

We have designed the tutorials to not simply be a repeat of work you did in lecture, but to give you different kinds of opportunities to problem-solve and practice what you've learned. We have three main goals for our tutorials:

1. Help you practice and review material covered in lectures and course readings.
2. Give you opportunities to *apply* and *extend* your computer science knowledge and skills to new domains and problems.
3. Foster a sense of community and build individual relationships among every one of you.

While attendance in tutorials is not graded, all of the material covered in tutorials is mandatory, and **attendance is strongly recommended**. Tutorials are a way for you to meet and work with other students in the course, and to receive individual attention from a member of the course staff in a more intimate setting than lecture. Like all synchronous components of the course, attending tutorials regularly is a way to make sure you keep on top of your learning in this course.

Note: we often try to give extra problems that we expect to take longer than the full tutorial time. Don't feel discouraged if you don't finish every part of every tutorial, as we give additional work to give you more practice to review and learn the material. Optionally, you may also use the tutorials to ask your TA questions about other coursework such as lecture worksheets and assignments, similar to office hours.

Assessments

You will complete three major kinds of assessments in this course: weekly preparation exercises, assignments, and tests/exams. **Note:** all weekly preparation exercises and assignments can be completed fully online. *The tests and exam will be held in-person on campus, with no exceptions.*

You will also complete two short surveys on *PythonTA*, which is one of the educational software tools we'll be using this semester. The following table summarizes the course assessments:

Assessment	Due Date/Date Held (all times are Eastern Time)	Weight
Weekly Preparation Exercises (Best 9 of 10)	Mondays before 11am	4.5%
Assignment 1	Friday, September 29 by 11:59pm	5%
Term Test 1	Monday, October 2 (during class)	5%
Assignment 2	Friday, October 13 by 11:59pm	10%
PythonTA Survey 1	End of October (exact date TBA)	0.5%
Term Test 2	Monday, October 30 (during class)	10%
Assignment 3	Friday, November 3 by 11:59pm	10%
Assignment 4	Friday, November 24 by 11:59pm	10%
Term Test 3	Monday, November 27 (during class)	10%
PythonTA Survey 2	Beginning of December (exact date TBA)	0.5%

Final Exam	TBD, held during the December Final Exam Period (https://www.artsci.utoronto.ca/current/faculty-registrar/exams-assessments/exam-assessment-schedule)	34.5%
------------	--	-------

Weekly Preparation Exercises (4.5%)

Weekly preparation exercises ("preps") consist of a few readings and short exercises that you complete before each week of lecture. We have designed these preps to help you stay on track and learn simpler concepts independently so that we can focus on more complex content and skills in lecture and tutorial.

Each prep consists of a short reading from the [CSC110/111 Course Notes](#) (<https://www.teach.cs.toronto.edu/~csc110y/fall/notes>), a series of short-answer comprehension questions hosted in an online **Quercus quiz**, and then some programming exercises that you will download and submit to using the online **MarkUs** application. Please note that:

1. You must submit all prep work individually. However, you may freely discuss all of your answers with your classmates, and with TAs and instructors on the course discussion board and during office hours / tutorials.
2. The prep deadline is firm, and no late submissions are accepted.
3. For Quercus comprehension quizzes, you can make as many attempts as you wish. Your score is shown after each attempt. If you submit multiple attempts, only your **best score** is used.
4. For the programming exercises, you may submit the required file(s) as many times as you wish on MarkUs. However, your submission is only graded after the deadline has passed, and only your **most recent submission** before the deadline will be graded.

Notes:

- The first prep exercise will be due **Monday September 11 before 11am**. It will be a shorter prep, and we'll be posting it after our first week of lectures on Friday September 8 – the readings for the prep will be assigned by Thursday September 7, so you have time to do the readings first before attempting the exercise.
- The final prep exercise will be due at the start of "Week 10" on **Monday November 20 before 11am**. There will be *no prep* on Weeks 11/12, to give you a bit of a break at the end of the semester!

Assignments (35%)

Assignments are larger pieces of work that span multiple course topics, and require you to apply and synthesize your knowledge and skills from multiple areas in computer science. They are a mixture of written and programming tasks, and will be graded for both correctness and clarity of communication.

Assignments will be posted online, and will be submitted to the MarkUs application. Assignments must be completed individually.

Assignments: Late Policy

There is a one-hour grace period after an assignment deadline, during which no penalty will be applied. Assignments submitted after this one-hour grace period are late and will be accepted only under the policy on special consideration and accommodations below.

Assignments: Special Consideration and Accommodations Policy

We recognize that unexpected problems, illness and disability-related barriers sometimes make it difficult to submit assignments on time. *(Note: Remember to value both your physical and mental health! We recognize that feeling emotionally unwell can be just as debilitating toward getting coursework completed on time.)* So, we are adopting a policy aiming to be as flexible as possible for a course of this size: You may request an extension of **up to one week** for one or more of the major assignment submissions by completing a form that will be made available on this page when the first assignment is released.

Just a note that if you do receive an extension this way, MarkUs may still continue to display the original deadline and it will appear as if your submission is late. After we grade your assignment, the penalty assigned by MarkUs will be waived. The maximum extension that can be allowed is one week. Any assignments submitted beyond the one week extension and the one-hour grace period (even 1 second beyond) will not be graded.

This policy is intended to cover students who are registered with Accessibility Services and require extra time to complete assignments as well as students who discover that they are unable to meet the original assignment deadline. Do not use it lightly to simply shift the original deadline. For example, if a student has been granted an extension of 1 week and then becomes ill on the extended deadline, **no further extension will be given** unless the student has been ill for more than 7 days or the student's college registrar is involved for extremely extenuating circumstances.

Note that this policy only applies to the four major assignments – **not** to weekly prep exercises which must be submitted on time. **No late submissions will be graded for prep exercises.**

Assignments: Doing your own work, on your own

The work you submit must be your own. It is an academic offence to copy the work of someone else. This includes their files, their words, and even their ideas. Whether you copy or let someone else copy, it is an offence. Academic offences are taken very seriously.

At the same time, we want you to benefit from working with other students. You are also welcome 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 assignment solutions is strictly forbidden*. The most certain way to protect yourself is not to discuss or share solutions with other students. Certainly, you must not let others see your solutions, even in draft form. Do not post your solutions on public online platforms like GitHub, as these can be searched and used by other students. (See [Your course work \(http://#your-course-work\)](http://#your-course-work) below for our advice about using private GitHub repositories.)

Please do not cheat. We want you to succeed and are here to help if you are having difficulty.

For more advice about maintaining academic integrity in CSC110 (including what forms of student collaboration *are* allowed!), please see [Academic Integrity in CSC110](#).

Term Tests (25%) and Final Exam (34.5%)

Tests are used to evaluate your learning in a focused setting periodically throughout the semester (term tests) and at the end of the course (final exam). Each term test will take place in-person, during a Monday lecture time, replacing the regular lecture.

IMPORTANT NOTE: You must receive a grade of at least **40%** on the final exam to pass CSC110. Students who do not meet this threshold (including students who do not write the final exam) will have their course grade lowered to no more than 48.

Technology requirements

To participate in this course, you must have reliable access to a personal computer to complete course work. A desktop computer or laptop are required; other computing devices, such as Chromebooks, tablets, and smartphones, are **NOT** sufficient to run the software required for this course.

We recommend bringing a laptop to lecture and tutorial, so that you can experiment with and complete various programming-related exercises. However, if you do not have access to a laptop you will still be able to participate and complete almost every exercise on paper, though you will be responsible for printing out exercise handouts and bringing them to class. (See below for information about accessing our department's on-campus computer labs.)

Software setup

You need to complete the CSC110 [Software Installation Guide](#) on your personal computer to make sure you have all the required software for this course. *Note:* we are using PyCharm to display, write, and run Python programs in this course. While we are not grading your use of PyCharm, if you choose to use a different program for Python programming, it will be your responsibility to translate instructions we give for using PyCharm, and your instructors and TAs may be unable to assist you. It is **highly**

recommended to use PyCharm for this course, even if you have previously used different software before.

The Department of Computer Science Teaching Labs

As first-year computer science students, you have access to our department's Teaching Lab rooms, located in the Bahen Centre, 40 St. George Street. These lab rooms are a popular study and work location for CS students, and have both computers and printers that you can access. For more information about the teaching labs, please check out [the CS Teaching Lab website \(https://www.teach.cs.toronto.edu/resources/introduction-for-new-students-using-the-teach-cs-labs/https://www.teach.cs.toronto.edu/resources/intro_for_new_students.html\)](https://www.teach.cs.toronto.edu/resources/introduction-for-new-students-using-the-teach-cs-labs/https://www.teach.cs.toronto.edu/resources/intro_for_new_students.html).

Textbooks and resources

There is no required textbook for this course. We'll be making use of a set of Course Notes that we have prepared for CSC110/CSC111, available for free online at <https://www.teach.cs.toronto.edu/~csc110y/fall/notes> (<https://www.teach.cs.toronto.edu/~csc110y/fall/notes>). Roughly half of these chapters will be assigned as prep throughout this term, and these are all required readings for the course.

Here are a few supplementary books and resources that you may useful for this course:

- ***Practical Programming*** (<https://pragprog.com/titles/gwpy3/>) by Paul Gries, Jennifer Campbell, and Jason Montojo.
 - ***How to Think Like a Computer Scientist*** (<http://www.openbookproject.net/thinkcs/python/english3e/>) by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers.
 - ***How to Prove It*** (https://librarysearch.library.utoronto.ca/discovery/search?query=any,contains,how%20to%20prove%20it%20daniel%20velleman&tab=Everything&search_scope=L) by Daniel J. Velleman.
 - ***Discrete Mathematics: An Open Introduction*** (<http://discrete.openmathbooks.org/dmoi3.html>) by Oscar Levin.
-

Accommodations and accessibility services

Students with diverse learning styles and needs are welcome in this course. The Accessibility Services staff are available by appointment to assess specific needs, provide referrals, and arrange appropriate accommodations. The sooner you let them and us know your needs, the quicker we can assist you in achieving your learning goals in this course. For more information on services and resources available

to students, including registering for accommodations, please see the U of T Accessibility Services website: <https://www.studentlife.utoronto.ca/as> (<https://www.studentlife.utoronto.ca/as>).

if you have a disability or health consideration that may require accommodations, please visit <http://www.accessibility.utoronto.ca> (<http://www.accessibility.utoronto.ca/>) **as soon as possible**.

Students who require accommodations for the term tests **need to register with Test & Exam Services**.

Special consideration for term tests

Students experiencing illness or other emergencies that prevent them from being able to complete homework on time, or write a term test, can request special consideration. To do so, complete the Special Consideration Request Form [↓](#) (coming soon) and email it to the course account (csc110-2023-09@cs.toronto.edu (<mailto:csc110-2023-09@cs.toronto.edu>)) from your U of T email address. You will receive an email response to your request within 1-2 business days.

IMPORTANT: *Submit your request soon as possible if you find yourself in such a situation. It is easier to resolve situations earlier rather than later.* If your emergency will affect your ability to complete coursework for more than a few days, or in multiple courses, we recommend you also talk to your registrar. You should also complete the absence declaration form on ACORN.

Special consideration for other homework

The flexible extension policy for assignment deadlines should cover all illness, disability-related barriers, and other special considerations for Assignments. A student who has been ill **for the entire 7 days between the assignment deadline and the extension date**, may contact us through the course email address.

As for the weekly prep exercises: the reason that we allow students to count the best 9 of 10 exercises is so that they can miss an exercise due to illness or other unexpected circumstances. Students who are ill for **more than one prep exercise**, can email the course email (csc110-2023-09@cs.toronto.edu (<mailto:csc110-2023-09@cs.toronto.edu>)) to request special consideration on the weighting of their completed work. Special consideration will not be granted for students who are only ill for a single prepare or perform exercise.

Remark requests

Mistakes sometimes happen when marking. If you feel there is an issue with the marking of an assignment or test, you may request that it be remarked.

For assignments: request a remark on MarkUs for the assignment you feel there was a mistake in marking for. You must give a specific reason for the request, referring to a possible error or omission by the marker. Remark requests without a specific reason will not be accepted.

For tests: please see the announcement about the test result availability for details once test marks are released.

For prompt turnaround, remark requests must be received within **one week** of when the item was returned.

Please note that when we receive a remark request, we may regrade the entire submission, not just a specific question. Your mark may go up or down as a result of the remark.

CSC110 Community Code of Conduct

[This section is based in part on the [Community Covenant \(https://www.contributor-covenant.org/\)](https://www.contributor-covenant.org/).]

All members of the course staff and all students are part of the same CSC110 community, and we share the common goal of creating a safe and positive learning environment for every student. Each of us is responsible for creating this environment, and must follow the guidelines below when participating in this course.

1. *Use welcoming and inclusive language. Show empathy towards other community members.*

Call people by their preferred names and pronouns. Do not make offensive comments about an individual or group (e.g., gender, sexual orientation, disability and mental illness, or race). Avoid humour or sarcastic remarks based on such comments or stereotypes.

2. *Be respectful of differing viewpoints and experiences. Gracefully give and accept constructive criticism.*

Look for (and reflect on) ideas and perspectives that are different than your own. Make a genuine effort to thank those who share them. It is natural to disagree with something a member of our community has written, and you are permitted to voice your disagreement. However, when doing so take the following into consideration: try to understand where the other person might be coming from; do not assume the other person's motives or draw inferences from their identity; be polite in your response and state where you agree.

3. *Be professional in your conversations.*

While conversations about topics unrelated to CSC110 or even the University of Toronto are certainly permitted (and encouraged), keep these conversations professional as you would in the workplace. Do not share sexual or violent content and avoid profanity.

4. *Respect the personal boundaries of each community member.*

While we encourage you to make use of this course's online platforms to meet each other to form academic and social connections, no one is obligated to do so. Everyone will have different boundaries and comfort levels that may change over time and depending on the situation. When in doubt, ask. If someone has asked you to respect one of their boundaries (e.g., not to contact them), with or without a reason, please respect their wishes. Do not reveal any person's personal information or private communications to a third person (or publicly) without receiving their explicit consent.

If you experience a violation of this code of conduct in a CSC110 space, or witness such a violation (even if it is not directed at you), or have any other concerns, please contact the course staff at csc110-2023-09@cs.toronto.edu (<mailto:csc110-2023-09@cs.toronto.edu>). We will respond to you in a timely manner and everything you say will be confidential.

Copyright notice

Course materials prepared by the instructor are considered by the University to be an instructor's intellectual property covered by the Copyright Act, RSC 1985, c C-42. These materials are made available to you for your personal use, and cannot be shared outside of the class or published (made publicly available) in any way. Posting course materials or any recordings you may make to other websites without the express permission of the instructor will constitute copyright infringement.

This notice applies to all course materials, including (but not limited to): course notes, lecture slides, lecture recordings, lecture and tutorial handouts, sample solutions, and assessment handouts, starter code, and solutions.

Lecture and tutorial recordings

You may not make your own recordings of video, audio, or text chat, of lectures or tutorials.

Your course work

Work that you complete for CSC110 (including exercises, assignments, and tests) may not be shared with other students or published, with one major exception (see below). This policy is to both protect the intellectual property of course staff (including, for example, the design and starter files for assignments), and to protect you from committing acts of academic dishonesty. For more information

on this topic, see [the Department of Computer Science website](#)

(<https://web.cs.toronto.edu/undergraduate/portfolio-advice>).

[GitHub \(https://www.github.com\)](https://www.github.com) is a popular option for computer science students and professionals to both collaborate in teams and publish their work online, including to develop a portfolio for potential employers. As we said in the *Academic Integrity* section, you should not put your work publicly on GitHub. However, you may use GitHub's **private repositories** to store your own work. (See [GitHub's instructions for creating a repository \(https://docs.github.com/en/github/getting-started-with-github/create-a-repo\)](https://docs.github.com/en/github/getting-started-with-github/create-a-repo) and select "Private" in Step 4.)

On the Use of Generative AI

Generative AI tools, such as ChatGPT and GitHub CoPilot, are strongly discouraged in CSC110. Here's why:

- One of the **most important goals** of CSC110 is for you to be able to **read and write code**.
- Use of these tools can give you a false sense of mastery of the course material. Grades are intended to give you some feedback on your understanding of the course material. These tools may help you earn higher grades than your actual level of understanding.
- Students who make use of generative AI to solve unsupervised coursework (e.g. Prep exercises, Assignments) are not developing the programming skills necessary to be successful on both supervised coursework (e.g. Term Test, Final Exam) and future courses (e.g. CSC111 and any other course that depends on mastery of the CSC110 material).
- It is an academic offence to submit work that is not your own without proper attribution, whether that is generated by another human, or by AI.
- Even if you are only using generative AI for studying and practice exercises that are not for marks, [it can sometimes give incorrect answers or explanations \(https://flyingbisons.com/blog/hallucinations-of-chatgpt-4-even-the-most-powerful-tool-has-a-weakness\)](https://flyingbisons.com/blog/hallucinations-of-chatgpt-4-even-the-most-powerful-tool-has-a-weakness), even on relatively simple programming questions.

Switching into CSC108

You will be able to switch from CSC110 to CSC108 up until the end of September. Students who are thinking about switching into CSC108 should speak with Sadia, who is the First-Year Computer Science Faculty Advisor. Please note that if you switch to CSC108, you will also be removed from CSC111 in the winter semester, but will be given the option of taking CSC148 and CSC165 instead. You will lose the Computer Science Admission Guarantee but will be able to apply to a computer science program after completing CSC148 and CSC165.