

Course Syllabus

Teaching Team

Instructor: [Jen Campbell](https://jencampbell.github.io/) ↗ (<https://jencampbell.github.io/>)

- Email: csc271-2026-01@cs.toronto.edu (<mailto:csc271-2026-01@cs.toronto.edu>)
- Office hours: Mondays 11am-12pm in BA 4268 (Bahen Centre), starting the week of January 12

Teaching Assistants:

- Amy, Cameron, Erica, Kate

Course Schedule

	Day	Time	Location
Tutorial	Tuesday	1:10-2:00pm	To Be Announced
Lecture	Wednesday, Friday	1:10-2:00pm	BA 1130

Course Description

Computational tools and methods are a cornerstone of the data scientist's toolbox, useful in a variety of applications and disciplines. This course builds on introductory data science and computer programming skills to equip students with several of these tools and methods. Computational methods for gathering and storing data via web APIs or web scraping or other formats; data pre-processing methods useful in data science algorithms; using version control and other tools to implement reproducible data science workflows; using web tools to communicate data science results and build data science products; creating, distributing, and accessing open-source data science software libraries. This course assumes prerequisite experience in computer programming, but does not require any additional knowledge or prior experience with any of the tools or methods covered.

By the end of this course, you should be able to:

- Understand the structure of and be able to navigate a file system.
- Install software packages and run programs from the command line.
- Recognize the stages of the data science lifecycle and write computer programs in Python to complete data science tasks, including data collection, cleaning, manipulation, analysis, and visualization.

- Use version control (git) to manage changes in a project and understand the purpose of version control.
- Understand what regular expressions are, when they are useful in data science, and how to construct and interpret regular expressions.
- Understand what web scraping is, know how to assess whether it is appropriate to scrape data from websites, and write computer programs in Python to scrape websites.
- Understand what a Web API is, how to evaluate third-party APIs, and write computer programs in Python to retrieve data using Web APIs.
- Understand the fundamentals of relational databases and write computer programs in Python using SQL to create, populate, and query databases.
- Produce reports and webpages using web tools.
- Create and distribute software packages.

Marking Scheme

	Weight	Date
Weekly Lecture Preparation	5%	Wednesdays by 1:00pm (before the first lecture of the week). *Best 8 of 10.
Weekly Tutorials	10%	Weekly tutorial Tuesdays 1:10-2:00pm and submission by Tuesday 10:00pm. *Best 8 of 10.
Assignment 1	10%	Monday Feb 2 by 10:00pm
Assignment 2	10%	Monday Feb 23 by 10:00pm
Midterm Test	10%	Wednesday March 4 1:10-2:00pm **TO BE CONFIRMED**
Project Proposal	5%	Monday March 9 by 10:00pm
Project	15%	Thursday April 2 by 10:00pm
Final Exam	35%	Scheduled by the Faculty of Arts & Science. To pass the course, students must earn at least 40% on the final exam.

*We drop the lowest two marks from your weekly lecture preparations and from your weekly tutorials in case you have to miss due to illness or other personal circumstances. See the section on Special Consideration for more details.

- Weekly Lecture Preparation

To prepare you for lecture, each week you will be assigned readings and a lecture preparation quiz based on the readings (a Quercus quiz) that is due before the Wednesday lecture. You have an unlimited number of attempts for each quiz and only the highest quiz score will be counted.

- Weekly Tutorials

There will be a 10 tutorials during the course (no tutorial during week 1 and no tutorial during one other week later in the semester with the exact date to be determined). Tutorials are designed to help you review and practice the course material.

You'll complete a few small tasks in the tutorial and submit your tutorial work **during the tutorial session** (or by 10:00pm on Tuesday). The grade will be based on attendance at the tutorial **and** an evaluation of the submitted work.

- Assignments

You will complete two assignments, which will give you practice applying concepts covered in the course.

- Midterm Test

The midterm test will be held during regular class time. More details about the midterm will be posted closer to the date.

- Project

The final project will involve using data science principles and methods to explore a problem and communicate the results.

- Final Exam

The final exam will take place in person during the final exam period. It is scheduled and administered by the Faculty of Arts and Science. More details on the final exam will come later in the term.

Course Policies

Accessibility Needs


The University of Toronto is committed to accessibility. If you require accommodations or have any accessibility concerns, please visit <http://www.accessibility.utoronto.ca> (<http://www.accessibility.utoronto.ca/>) as soon as possible.

Students who require accommodations for the midterm and/or the final exam must register with Accommodated Testing Services.

Late/missed work, special consideration, and accommodations

We recognize that unexpected problems, illness, and disability-related barriers sometimes make it difficult to submit coursework on time. Our course policies have been designed with that in mind.

- For Weekly Lecture Preparation and Weekly Tutorials
 - The best 8 of 10 policy for the exercises and tutorials accounts for missing exercises and/or tutorials due to late enrolment, illness, or other unexpected circumstances.
 - No late lecture preparations will be accepted.
 - For tutorials, there is a one-hour grace period after each tutorial deadline during which no penalty applies. No late tutorial submissions beyond that one-hour grace period will be accepted.
- For Assignments
 - There is a one-hour grace period after each assignment deadline during which no penalty will be applied.
 - We will grant extensions of up to 3 days without penalty for assignment submissions submission. 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. You may make use of this policy on as many assignments as you require.

[CSC271 Assignment Special Consideration and Accommodations - Winter 2026](https://forms.office.com/r/kfvwD57dh4) 
(<https://forms.office.com/r/kfvwD57dh4>)

- When an extension is requested under this policy:
 - the extension begins from the original assignment/project due date and time
 - the deadline is typically updated on MarkUs within about an hour of submitting the form
 - the maximum extension allowed is 3 days (plus the one-hour grace period). Any assignments submitted beyond the 3 days + 1 hour will not be graded.
 - Requests for longer extensions of up to 7 days from the original assignment deadline will only be considered with appropriate documentation. Contact csc271-2026-01@cs.toronto.edu (<mailto:csc271-2026-01@cs.toronto.edu>)
 - Extensions beyond 7 days are rare and will only be considered for extremely extenuating circumstances. These longer extension requests require the involvement of your college registrar or accessibility advisor. Contact csc271-2026-01@cs.toronto.edu (<mailto:csc271-2026-01@cs.toronto.edu>)
- For the Project Proposal and Final Project Submission
 - There is a one-hour grace period after each project deadline during which no penalty will be applied.
 - Requests for late submission will be considered on a case by case basis with proper documentation. Contact csc271-2026-01@cs.toronto.edu. (<mailto:csc271-2026-01@cs.toronto.edu>)

Academic Integrity

You are responsible for knowing the content of the [University of Toronto's Code of Behaviour on Academic Matters](http://www.governingcouncil.utoronto.ca/policies/behaveac.htm) (<http://www.governingcouncil.utoronto.ca/policies/behaveac.htm>).

As a general rule, we encourage you to discuss course material with each other and ask others for advice. However, it is not permitted to share complete solutions or to directly share code for anything that is to be handed in. For example, "What Python function did you use for that tutorial question?" is a fair question; "Please show me your Python code for that tutorial question." is not.

If you have any questions about what is or is not permitted in this course, please do not hesitate to contact your instructor. Please also see the course policy on the use of Generative AI tools.

Use of Generative AI (e.g., ChatGPT or CoPilot)

In this course, you may use generative artificial intelligence (AI) tools, including ChatGPT, Microsoft Copilot, and GitHub Copilot, as learning aids and to help complete tutorials, assignments, and the final project. However, you must be able to clearly and easily explain and justify the work submitted. Submitting coursework without understanding it fully may be considered plagiarism and penalized accordingly. You are not permitted to use generative AI on the midterm test or final exam.

We caution you to not rely too heavily on these tools to complete your coursework. Instead, we recommend treating generative AI as a supplementary tool only for exploration or drafting content. Ultimately, you (and not any AI tool) are responsible for your own learning in this course, and for all the work you submit for credit. It is your responsibility to critically evaluate the content generated, and to regularly assess your own learning independent of generative AI tools. Over reliance on generative AI may give you a false sense of how much you've actually learned, which can lead to poor performance on the midterm test or final exam, in later courses, or in future work or studies after graduation.

The use of any generative AI tools in your assignments and final project must be documented and that documentation must be submitted. The documentation should include what tool(s) were used, how they were used, and how the results from the AI were incorporated into the submitted work. The course instructor reserves the right to ask students to explain their process for creating their project.

Generative AI is not required to complete any aspect of this course.

Technical Issues

- No extensions or special consideration will be granted to individual students on exercises, tutorials, assignments, or projects for reasons of technical difficulties.
- To ensure that you are not affected by a technical difficulty on your end, we strongly recommend that you: (1) give yourself extra time in advance of the deadlines in case you have trouble

submitting, and (2) regularly back up your assignment files, such as by submitting them to MarkUs (yes, you can submit your work, including incomplete work, multiple times!), by emailing the assignment file to yourself, or by using some other method of backup. Make sure whatever backup method you choose is not public.

- In the unlikely event of a MarkUs or Quercus outage affecting your ability to submit your exercises or assignments, we will post an announcement with instructions on how to proceed.

Equity, Diversity, and Inclusion

The University of Toronto is committed to equity, human rights and respect for diversity. All members of the learning environment in this course should strive to create an atmosphere of mutual respect where all members of our community can express themselves, engage with each other, and respect one another's differences. U of T does not condone discrimination or harassment against any persons or communities.