Chandra Gummaluru
August 02, 2025

CSC263 Fall 2025 Syllabus

> The details in this document are preliminary and subject to change until **September 15, 2025**. After this date, the details will be finalized and changes will require a vote from the class.

Course Information

The course is **CSC263: Data Structures and Analysis** at the University of Toronto. You can find more information about the course here. Below is the general syllabus for Fall 2025.

- **Course Website:** All course material will be posted on this page.
- **Lecture Sections:**

Section	Tutorial	Lecture A	Lecture B
Section 1	Mondays 13:00 - 14:00 KP108	Wednesdays 13:00 - 14:00 RW117	Fridays 13:00 - 14:00 MP102
Section 2	Mondays 15:00 - 16:00 RW110	Wednesdays 15:00 - 16:00 RW110	Fridays 15:00 - 16:00 MP203

- **Communications:**
  - All course-related communication should take place on the course's Piazza page.
  - Please do NOT email the teaching team about course-related matters. We receive a large volume of emails, which makes it difficult to track and respond consistently. To ensure fairness and efficiency, we will not be answering course-related questions by email.

Learning Objectives

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

1. Understand and implement fundamental data structures and algorithms.
2. Analyze algorithmic efficiency, including worst-case and amortized performance.
3. Apply data structures and algorithms to solve real-world problems.

Teaching Team

Instructors

There is one (1) course instructor (CI) for this course.

Chandra Gummaluru
Assistant Professor, Computer Science

Office: BA4201A
Website: chandra.gummaluru@github.io
Email: chandra@cs.toronto.edu (for personal matters only)
Office Hours:

- TBD — This time is reserved for course-related questions (assignments, concepts, exam prep, etc.).

- Fridays, 14:00–15:00, Walking Office Hours — Meet at BA lobby at

14:10. This time can be used for course questions or for broader discussions, such as research, career paths, or job advice.

Teaching Assistants

There are nine (9) teaching assistants (TAs) for this course.

- Jacob Gray
- Yibin Zhao
- Warren Zhu
- Xing Hu
- Rishabh Kothary
- Victoria Cimpu
- Jacob Mausberg
- Ningning (Mila) Liu
- Amy Lai

Lecture Schedule

All lectures and tutorials are offered in **both sections**. Quizzes take place during tutorials.

All dates are **tentative** and subject to change. Extra time has been reserved at the end of the course to account for needing to use a slower pace than originally planned, revisiting previously covered material, or students asking many questions (which is a good thing). If this happens, the schedule (and quiz dates) will be adjusted accordingly. Quizzes will never be moved to an earlier date (only later). You will be given at least one (1) week advance notice if a quiz date is changing. If extra time remains after all required material has been covered, the remaining few lectures will be used to introduce additional data structures. These topics will not be tested but are intended to support your preparation for future courses in data structures.

Week	Monday (Tutorial)	Wednesday (Lecture)	Friday (Lecture)
1	—	Syllabus & Prologue	Complexity Analysis

2	—	Graphs	Shortest Path Problems: BFS
3	Tutorial	Shortest Path Problems: DFS	Shortest Path Problems: UCS
4	Quiz 1	Smallest Connected Network Problems: Kruskal's Algorithm	Union-Find
5	Tutorial	Smallest Connected Network Problems: Prim's Algorithm	Guest Lecture 1
6	Tutorial	Unsorted Dictionaries (Part 1)	Unsorted Dictionaries (Part 2)
7	Tutorial	Sorted Dictionaries (Part 1)	Sorted Dictionaries (Part 2)
8	Quiz 2	Queues (Part 1)	Queues (Part 2)
9	—	—	—
10	Tutorial	Amortized Analysis	Guest Lecture 2
11	Tutorial	TBD	TBD
12	Quiz 3	TBD	Guest Lecture 3
13	Tutorial	Review / Wrap-up	Review / Wrap-up

Evaluations

There are three (3) types of evaluations in this course.

Assignments

Each assignment involves writing some code in Python which will be graded using an auto-grader.

- **Weight:** 3 × 10% each
- **Duration:** 14 days
- **Dates & Topics:**

  - **Assignment 1:** 17/09 – TBD
  - **Assignment 2:** 14/10 – TBD
  - **Assignment 3:** 14/11 – TBD

- **Policies:**

  - Assignments are released at 00:00:01 are due at 23:59:59 on the 14th day after (including the release date).
  - You will be given five (5) grace tokens, which can be used across all assignments in any fashion. Each grace day grants you an extra 24 hours to submit an assignment beyond the due date.
  - For every 24 hours that an assignment is late (not including grace tokens), 20% will be subtracted from your earned grade, when expressed as a percentage.
  - Your actual grade for the assignment, expressed as a percetnage, can be determined using the following formula:

    $$\max\left(0, x - 20(\lceil y/24 \rceil - z)\right)$$

    where x is the grade you earned expressed as a percentage, y is the number of hours late, and z is the number of grace days used.

  - If you cannot submit an assignment due to a valid reason as defined below, its weight will be transferred to the final exam.

Quizzes

- **Weight:** 3 × 10% each

- **Duration:** 50 minutes

- **Dates & Topics:**

  - **Quiz 1:** 22/09 – Time Complexity, Graphs, Shortest Path Problems
  - **Quiz 2:** 20/10 – Smallest Connected Network Problems, Unsorted Dictionaries
  - **Quiz 3:** 17/11 – Sorted Dictionaries, Queues

- **Policies:**
    - If you cannot write a quiz due to a valid reason as defined below, its weight will be transferred to the final exam.

Final Exam

- **Weight:** 40%
- **Duration:** 180 minutes
- **Topics Covered:**
    - Time Complexity
    - Graphs
    - Shortest Path Problems
    - Smallest Connected Network Problems
    - Dictionaries
    - Queues
    - Amortized Analysis
- **Policies:**
    - You must obtain **at least 40%** on the final exam to pass the course.

Tips

This course is designed around a cycle of preparation, engagement, review, and practice. Each step builds on the previous one to help you understand the material deeply rather than just memorize it.

1. Prime (Before Lecture)

- Skim the lecture slides ahead of time.
- Don’t worry about details — just get a sense of the main ideas and the “lay of the land.”

2. Participate (During Lecture)

- Lectures are partly instructional, partly conversational.
- Focus on engaging in discussion and asking questions rather than copying everything down.
- Annotate slides only with what’s unclear or what you might forget later.

3. Process (After Lecture)

- Review the material on your own over the weekend.
- Revisit the slides and notes, filling in any gaps in understanding.

4. Practice (Before Tutorial)

- Work through the weekly worksheet.
- The worksheets are **not graded**, but they are essential for learning.
- Make a genuine attempt before tutorial so you can identify what you don’t yet understand.

5. Prove (In Tutorial)

- Bring your completed worksheet to tutorial.
- Go over solutions with the TA, verify your reasoning, and clear up any remaining doubts.

By following this cycle each week, you’ll not only keep up with the course but also build lasting understanding of the material.

Academic Integrity

Your work in this course should reflect your own understanding. To keep expectations clear, here is the policy for different types of evaluations:

Supervised Evaluations

Quizzes and exams

- Must be completed independently with no outside help.
- No external resources (AI tools, websites, peers, etc.) are permitted.
- All submitted answers must be entirely your own.

Unsupervised Evaluations

Worksheets and auto-graded assignments

- You may use any resources (textbooks, online references, AI tools, etc.).
- You must explicitly acknowledge all resources used. For example:

  > “I used ChatGPT to clarify the concept of convexity”
  > “I referred to GeeksforGeeks for syntax of a Python function”

- Assignments are formative: they help you practice and prepare for supervised assessments. Over-reliance on external tools may make supervised assessments more challenging, so be careful.

Guidelines

You are encouraged to engage in conceptual discussions with classmates, utilize online resources, and employ AI tools to brainstorm or clarify ideas. It is also acceptable to use online tools or AI to assist with small portions of an assignment. The guiding principle is that all submitted work must reflect your own understanding, and directly copying solutions from others or from AI is strictly prohibited.

A useful guideline is: if you cannot explain a solution independently—without referring to another student, an AI tool, or online resource—it likely does not meet the standards of academic integrity.

Specifically:

- ✓ Consulting a classmate for explanations or guidance is permitted.
- ✓ Using AI to generate small snippets of code or to clarify a concept is permitted.
- ✗ Copying another student’s solution is prohibited.
- ✗ Using AI to produce full solutions or entire assignments is prohibited.

In cases of suspected academic integrity violations, the teaching team may conduct an interview to assess your understanding of the material, using the principle above. Confirmed violations will be addressed in accordance with University of Toronto academic integrity policies. For additional strategies and guidance, see Academic Integrity at U of T.