

CSCC24H3 Y

Principles of Programming Languages

Summer 2025 Syllabus

Course Meetings

CSCC24H3 Y

Section	Day & Time	Delivery Mode & Location
LEC01	Thursday, 12:00 PM - 2:00 PM	In Person: IA 2040
TUT0001	Wednesday, 3:00 PM - 4:00 PM	In Person: IA 3010
TUT0002	Tuesday, 11:00 AM - 12:00 PM	In Person: IA 3010

Refer to ACORN for the most up-to-date information about the location of the course meetings.

No tutorials in the first week.

Course Contacts

Course Website: <https://www.cs.utoronto.ca/~trebla/CSCC24-2025-Summer/>

Instructor: Albert Lai

Office Hours and Location: Tuesday and Friday 14:00-15:30, IA 3180 (CS Help Centre)

Course Overview

Major topics in the design, definition, analysis, and implementation of modern programming languages. Study of programming paradigms: procedural (e.g., C, Java, Python), functional (e.g., Scheme, ML, Haskell) and logic programming (e.g., Prolog, Mercury).

Course Learning Outcomes

Demonstratable, applicable understanding of basic topics in programming languages and associated theory.

Prerequisites: CSCB07H3 and CSCB09H3 and [CGPA 3.5 or enrolment in a CSC Subject POST]

Corequisites: None

Exclusions: CSC324H

Recommended Preparation: None

Credit Value: 0.5

The nominal prerequisite is minimal to give flexibility to excellent students. This course will

proceed as if students have completed all B-level CSC, MAT, and STA courses in CS/SE POSTs.

Course Materials

Required, recommended, and reference reading are available online and listed on the course website alongside the list of lectures.

Required websites to check:

- The course website.
- Quercus: Announcements, forum, solutions, past tests and exams.
- [MarkUs](#): Handing in and returning assignments.
- Gradescope: Returning tests.

Marking Scheme

Assessment	Percent	Details
Midterm Test	16%	1.5 hours. 1 aid sheet allowed (letter or A4 size, 2 sides).
Assignments	40%	4 assignments.
Tutorial exercises	10%	11 tutorial exercises. The best 10 are chosen.
Final Exam	34%	3 hours. 2 aid sheets allowed (letter or A4 size, 2 sides). A final exam mark of at least 40% is required to pass the course.

Late Assessment Submissions Policy

While I would like to be accomodating when there is a legitmate reason, you must discuss with me and obtain my agreement on an alternative arrangement. This includes Absence Declaration—until you initiate a discussion, I will do nothing and make no change.

If the reason is foreseeable, you must obtain my agreement in advance.

I honour requests from AccessAbility.

When you don't have a good case, there are 4 grace credits for the whole term at your disposal. You may use at most 2 grace credits per assignment. This is automated on MarkUs, so just submit late when you need to. These cannot be used for tutorial exercises.

Policies & Statements

Academic Integrity

The University treats cases of cheating and plagiarism very seriously. The University of Toronto's Code of Behaviour on Academic Matters (<http://www.governingcouncil.utoronto.ca/policies/behaveac.htm>) outlines the behaviours that constitute academic dishonesty and the processes for addressing academic offences.

Potential offences in papers and assignments include using someone else's ideas or words without appropriate acknowledgement, submitting your own work in more than one course without the permission of the instructor, making up sources or facts, obtaining or providing unauthorized assistance on any assignment.

On tests and exams, cheating includes using or possessing unauthorized aids, looking at someone else's answers during an exam or test, misrepresenting your identity, or falsifying or altering any documentation required by the University.

Accommodations

Students with diverse learning styles and needs are welcome in this course. In particular, if you have a disability/health consideration that may require accommodations, please feel free to approach me and/or the AccessAbility Services Office as soon as possible.

AccessAbility Services staff (located in Rm IA5105, Sam Ibrahim Building) are available by appointment to assess specific needs, provide referrals and arrange appropriate accommodations 416-287-7560 or email ability.utsc@utoronto.ca. The sooner you let us know your needs the quicker we can assist you in achieving your learning goals in this course.

Generative Artificial Intelligence Tools

Students must submit only work of their own; the use of generative artificial intelligence tools, including ChatGPT and other AI writing and coding assistants, for the completion of, or to support the completion of, an examination, term test, assignment, or any other form of academic assessment, may be considered an academic offense in this course. However, students may use artificial intelligence tools in this course as learning aids.

Maturity Requirement

Normally CMS instructors are supportive and lenient, in good faith that students are diligent in return (co-op thinking: they do more, you do more); inevitably, some students misunderstood it (tug-war thinking: they do more, you do less). And since the Internet (humans or otherwise) is so eager to explain everything, students have forgotten that learning CS consists of a large part of practicing, and only a small part of receiving guidance. So I need to be strict before I can be collegial:

This is a 3rd-year course, requiring a commensurate level of intellectual maturity, independence, reading comprehension, due diligence, and sense of responsibility. Quality of my answers to your questions is proportionate to quality of your questions.

Furthermore, this course proceeds as if you have completed all B-level CSC, MAT, and STA courses in CS/SE POSTs. Assessments that assume mastery of said courses will be considered fair.

Anti-Piazza position statement: Piazza encourages instant gratification, high-frequency spoon feeding, and fishing for answers. Those are anti-theses to learning. I will use a much simpler course forum, where possibly no single post is “the answer”, and you will have to study the whole conversation and think for yourself.

Re-Grades

Re-grades are for instructors' and graders' errors, not for yours. Especially not the kind you can easily check right before or even after submission, such as: wrong files, wrong filenames, lateness by a few seconds or minutes, (for program code) compile-time errors.

Leniency can still be offered, but only if you speak up ASAP.