Course Information

CSC 324 Principles of Programming Languages

University of Toronto

Dept. of Computer Science

Winter Semester, 2023

Course Description

In this course, we will learn about programming paradigms that are quite different from the imperative programming style taught in other introductory programming courses. We will cover major topics in the development of modern programming languages including syntax and semantics specification, type systems, type inference, exception handling, information hiding, structural recursion, and higher-order functions. We will study Racket and Haskell, two functional programming languages.

Prerequisites

The study of programming languages is unique in that it blends the practical skill of translating a problem into a solution that a computer can run, with theoretical underpinnings and formalism from the mathematics community.

CSC263 or CSC265 is required for this course. Specifically, this translates to you needing to have:

- Familiarity with common data structures such as linked lists and trees;
- Competency in some prior programming language, such that you could implement one of the above data structures;
- Comfort with recursion, and understanding and writing proofs by induction.

In essence, if you're reasonably convinced that you could design a binary tree datatype and write a recursive function/method that traverses it, and explain in your own words why your solution is correct, you will be in good shape for this class.

Learning Objectives

Upon successful completion of this course, you will learn the different concepts of programming beside imperative programming style. You will learn several fundamentals concepts of studying programming languages: formal syntax and semantics, recursion, higher-order functions, type systems. Also, you will learn how those concepts allow to ensure safe memory access, prove program correctness, and support good engineering practice (information hiding and abstraction).

Staff

LEC0101: Dr. Sidi Mohamed Beillahi, BA3242 (`sm.beillahi@utoronto.ca`). Office hours: Fri 3:20-4:30 pm or by appointment.

LEC0201: Prof. Fan Long, BA3250 (`fanl@cs.toronto.edu`). Office hours: Fri 3:00-4:00 pm or by appointment.

The Course Email: `csc324-2023-01@cs.toronto.edu` (Course email to be used for general inquiries like questions on material and remarking and to contact the teaching team).

Homeworks and assignments submission is through MarkUs (`https://markus.teach.cs.toronto.edu/2023-01/`).

The Course Piazza link: `https://piazza.com/utoronto.ca/winter2023/csc324` (Piazza page to be used for questions on the homeworks and assignments. Announcements will be made on Quercus).

Lecture Schedule

LEC0101 (Dr. Beillahi): Mon (1pm-2pm, LM161), Wed (1pm-2pm, BA3185 & BA3195), and Fri (1pm-2pm, LM161).

LEC0201 (Prof. Long): Mon (2pm-3pm, MP102), Wed (2pm-3pm, BA3185 & BA3195), and Fri (2pm-3pm, MP102).

Wednesday lectures, with few exceptions TBD, are reserved to tutorials and labs to practice notions seen in regular lectures. In labs, we practice solving programming exercices, and questions will be answered about the lecture material. If you would like your TA to go over specific topics, please let them know in advance via the course email so they can prepare (please include the course code and the section in the subject).

Important Dates

	Due
Homework 1	Monday, January 30th, 5pm
Homework 2	Monday, February 13th, 5pm
Homework 3	Monday, February 20th, 5pm
Assignment 1	Monday, February 27th, 5pm
Midterm	Wednesday, March 1st, 1-3pm
Homework 4	Monday, March 13th, 5pm
Homework 5	Monday, March 27th, 5pm
Assignment 2	Monday, April 3rd, 5pm
Final	TBD

Textbook and Class Contents

There is no required textbook for this course. However, the slides are based on David Liu notes that you can find on Quercus.

In this class we will cover the following material:

- Lambda calculus and formal semantics
- Recursion
- Evaluation order and Lazy evaluation
- Metaprogramming and Macros
- Functional programming
- Determinism
- Type systems
- Polymorphism
- Functors and Monads

Course Webpage

All official announcements will be posted on Quercus (`q.utoronto.ca`). *It is your own responsibility to check it at regular intervals, i.e. once per day.* It is strongly recommend to select "Notify immediately" for course announcements in your account notification settings. This will forward all announcements to your university email. Questions on the material (*i.e.,* lectures, exams, etc) should not be posted on Quercus, rather should be asked during lecture and tutorial sessions or sent to the course email: `csc324-2023-01@cs.toronto.edu`.

On Quercus you will find homeworks and assignments, lecture slides, and other useful resources. All homeworks and assignments will be submitted via MarkUs.

Course Requirements and Grading Scheme

The final grade for this class has four components:

- **homeworks:** There will be five homeworks you will do individually. Homeworks are simple exercices to practice lecture material. More details about proper homework submission are on Quercus. Homeworks account for 20% of your grade (4% each).
- **assignments:** There will be two assignments you will do individually. More details about proper assignment submission are on Quercus. Assignments account for 30% of the grade (15% each).

  Each student will receive seven flex tokens to extend a deadline for a homework or an assignment by four hours.
- **midterm exam:** There will be one and a half hours. The midterm accounts for 18% of the grade.

- **final exam:** Two hours, and it accounts for 32% of the grade.

Exam dates and rooms will be announced at a later time. Exam material will be simpler versions of the problems you will see in the homeworks and assignments. Exam type material will be practiced regularly at tutorials and homeworks.

Remarking: You have exactly *seven* (7) working days to submit your work (homework, assignment, or midterm) for remarking from the time marks are released. To request a remark, state the problem number(s) submitted for remarking, the reason you believe you were marked unfairly, and your name and email address who should be contacted for clarifications in an email, attach a pdf of your homework and any TA feedback received, and submit to MarkUs or send to the course email (`csc324-2023-01@cs.toronto.edu`). If there is a legitimate reason for a late assignment or exam absence (illness, etc.), please email the course email. Keep in mind that official documentation (doctor letter, etc) is required for an exemption. When a homework, assignment, or midterm is submitted for remarking, the entire document may be regraded (not just the sections you believe were marked unfairly) and *it is possible you will receive a lower mark than before* if we determine the initial grader was too generous.

Cheating Policy

Cheating is against "fair–play" and will not be tolerated under any circumstances. While the pressures of many classes, homeworks, work and/or extracurricular activities can be great, this is *never* an excuse for copying solutions from others. **The University holds among its highest principles the notion of academic freedom and integrity. Cheaters will face the University's disciplinary committee as well as receive a failing grade in this course.** If you think that there is an issue that influences your performance in the class then talk to the instructor.

Racket

- Racket installation: `https://download.racket-lang.org/`
- Racket documentation: `https://docs.racket-lang.org/`
- Racket programming tutorials:
  - `https://docs.racket-lang.org/quick/index.html`
  - `https://docs.racket-lang.org/redex/tutorial.html`

Haskell

- Haskell installation: `https://www.haskell.org/downloads/`
- Haskell documentation: `https://www.haskell.org/documentation/`
- Haskell programming tutorial: `http://learnyouahaskell.com/`

Further Readings

- `Fundamental Concepts in Programming Languages`
- `Programming Languages: Application and Interpretation`

How to Get the Most out of this Course

Above we listed some materials that you can read to expand what is taught in the lectures. The internet is full of additional scientific and historical material. Read the course webpage regularly and email the TA's if you have any questions. Attend lectures, labs&tutorials and office hours. Keep up with the pace of the class. Make sure you understand the solutions to all homeworks questions.