COMP 3007 Fall 2025 Course Outline

Lectures

In person, Wednesdays and Fridays 08:35-09:55.

Course website

See Brightspace.

Course staff

Role	Name	Email
Instructor	Douglas Howe	douglashowe@cunet.carleton.ca
TAs	TBD	

Prerequisites

1. COMP 1805 and COMP 2401 with a minimum grade of C-.
2. COMP 2402, COMP 2404 and COMP 2406, or equivalents (see calendar).

THE THREE A'S!

This course has weekly **Assignments**. They're only worth a total of 10% of the course, but if you don't do them, and on your own, you will not succeed in the course.

The **Assessments** (quizzes and exam) all involve coding on your laptop and are tightly tied to the assignments, often sharing some of the same code. They are intensively monitored to ensure you do the coding unassisted: they are in-person with proctors and also use CoMaS (see below).

The use of **AI** assistants (or ”agents”) is, of course, strictly disallowed on assessments. However, we will not be checking for their use in assignments. AI can likely do each assignment with one prompt. The reason to do the assignments yourself is to learn the material, or, in more universally appealing terms, to succeed on the assessments and hence get a good grade in the course.

ChatGPt, Copilot, Claude et al are excellent as tutors. They can give you exercises, explain things, check your understanding etc. However, be aware of a trap that students sometimes fall into. They don't get ChatGPT to do the assignment, but they do directly enlist its help and get it to ”suggest” approaches, snippets etc. They get to a point where they feel they have a solution that they completely understand and then conclude they've learned enough. But understanding code is not the same as writing code! It's analogous to natural

languages. Reading/listening and writing/speaking are very different activities. E.g. your instructor can read French at a university level but is unable to carry on a conversation with a two-year old.

Course schedule

Assignments will be due end-of-day every Monday *except* during the fall break. Quizzes will be held after every three assignments.

Event	Date(s)
First lecture	09-03 (Wed)
Registration ends	09-16
Quiz 1	09-24 (Wed)
Quiz 2	10-15 (Wed)
Fall break	10-20 to 10-24
Quiz 3	11-12 (Wed)
Withdrawal deadline	11-15
Quiz 4 and end of course	12-01 (Wed)
Final exam period	12-08 to 12-20

Laptop Requirement

The BCS degree has a laptop requirement. All students in this course will need to bring their laptops to in-class quizzes and the final exam. Be sure your laptop can run for 80 minutes. Don't worry if the battery won't last for the 3-hour final exam. The site will have power for those that need it. However, quizzes are done in the class room and there is no power there.

Main course learning outcomes

- moderate proficiency using a modern functional programming language
- apply standard recursive techniques for problem solving
- design and apply appropriate inductively-defined data-types in recursive problem-solving
- explain the general practical differences between functional programming languages and ”imperative” languages (such as Javascript, Java and Python)
- solve simple exercises involving the lambda calculus and explain how it can serve as a theoretical foundation of functional languages
- apply appropriate abstractions in a functional language, especially using modern type-based abstraction mechanisms

- explain the functional analogs/counterparts of common imperative programming patterns/paradigms

Course structure

This is a *synchonous in-person* course. The instructor intends to record lectures and post them on the course website to assist students who are ill etc, but there is no guarantee. If for some reason a lecture does not get recorded, it will not be redone. Lectures might be streamed live, but this should *not* be counted on.

Lectures will be a mix of slides and ”live coding”. The slides and code will be posted on the course website along with a recording of the lecture (if available). Lecture attendance is optional (but see above about videos and streaming).

Extensive use will be made of ”Ed Discussion”, a Q&A tool being used by most of the top US universities. This will be the main place where you can ask questions and have them answered. Answers can be from course staff (instructor and TAs), or from other students. Answers by students may be endorsed/annotated/highlighted by staff. It is expected that most interaction outside of class will be done this way. Conventional 1-1 office hours will only only be for students who are struggling with the material enough that they are unable to formulate specific questions or have so many questions that an in-person meeting is more efficient.

All of the term work of the course will involve the Haskell programming language. See the ”Help” section of the course website for instructions on how to install Haskell.

Textbook and references

There is no textbook for the course, but the online book Learn you a Haskell for Great Good is an excellent tutorial-style book for learning basic Haskell. Relevant sections will be pointed out at the course proceeds. Other web resources may be specified during the course.

Discord vs Ed

There is no official Discord channel for the course. Students, as always, are free to use one, as long as they adhere to University behaviour standards, but the course staff will not be answering course-related questions there, and will not be correcting any misinformation appearing there. The Ed discussion tool (see above) is the place to get help with course material, and to discuss it with other students. Students are strongly encouraged to use Ed for all course-content questions. The answers will be reliable and will be available to all students in the class. Leave Discord for social stuff, griping about the instructor etc.

Assessment scheme

Weight	Course component
10%	Weekly assignments, lowest two dropped
45%	Best 3 of 4 quizzes, 15% each
45%	Final exam

Late assignments

Late assignments will not be accepted.

Missed assignments and quizzes

The dropping of low scores in the assessment scheme is intended to account for work that's missed for reasons beyond a student's control. There will be no other accommodations granted for missed quizzes or assignments *no matter what the reason*, so be careful to save your ”free passes” for illnesses or other events out of your control.

E-proctoring

Quizzes and the final exam will be done in-person using your laptop. You will write and test programs, and submit the code electronically (method TBD) .

The quizzes and the final exam will use CoMaS, an e-proctoring tool developed in Carleton's School of Computer Science and now officially supported by the university's Exam Services unit.

CoMaS is a very small tool you install on your laptop for the assessment. It does not look at anything that was on your laptop before the exam starts. During the exam, it takes occasional screen shots and webcam pictures, restricts network access, and monitors file changes. After the assessment the tool can be deleted.

The minimum system requirements for your laptop to run CoMaS are here. CoMaS *might* work with other linux distros, but it's a crapshoot. Even if you meet all the requirements be sure to try out CoMaS on your device *well before* the assessment so you have time to get help from the CoMaS support team. Course staff cannot help with CoMaS issues.

CoMaS has been thoroughly vetted by the university. If you're concerned about privacy, maybe the following will help.

1. You almost certainly have already installed apps, needing similar access, from sources far less trustworthy than a large public university, including sources whose business model relies on harvesting personal information.

2. A breach of student privacy using CoMaS, if it became public, would be an existential crisis for the university. One thing I've learned about senior academic management is that if anything gives them nightmares it's the possibility of negative national press.

If you have questions about CoMaS and privacy, please see the CoMaS Privacy FAQ.

If for some reason you can't tolerate temporarily installing CoMaS, the only alternative is to write the test on paper. You wouldn't get the help of a programming environment, or running tests to debug your programs, but it would still be a reasonably fair test since the programs you will be asked to write will be tiny and marks will not be deducted for superficial syntactic errors.

University Policies

All courses at Carleton follow the policies/procedures below.

- Academic Integrity
- Student Rights and Responsibilities
- Academic Accommodations