Carleton University, Computer Science, Winter 2026

Comp 3002 Compiler Construction

Lectures/Instructor: Thursday 6-9pm, **W.R. Lalonde**, wilf.lalonde@gmail.com

Room location: *log in to Carleton Central.*

Weekly Assignments: Due Wednesday in Brightspace at Midnight (for assignment provided the week before). Assignments not done are worth 0; late assignments are subject to a penalty of 5 marks. Last day for submission of all assignments (late or not) is the last day of the term.

TAs: **Semantha Headington, Zahra Mousavi**

Course calendar description: **Compiler Construction** The structure, organization and design of the phases of a compiler are considered: lexical translators, syntactical translators, scope handlers, type checkers, code generators and optimizers. Components of a compiler will be implemented.
Prerequisite(s): COMP 2402

Course Goals: To develop the tools needed to write a compiler for a simple language and to write a compiler using those tools. Most assignments make use of table driven scanners and parsers.

Help Labs (voluntary but recommended): **Sat** 9am (4th year labs 4155, as long as you want; meet in the Herzberg tunnel to be allowed in since the buildings are locked on weekends). The idea is to work on the actual assignments with help from me, TAs, and other students in the course. Free extra 5 marks per lab session for the first 3 sessions. If there is demand, we can schedule additional weekday labs.

Basic Idea: On Saturday (you get help from myself and TAs in an optional lab where we typically finish the assignment giving you plenty of time to work on other courses during the week). We can help you ascertain whether your work is correct. The expectation is that the lab (because we can help) is the only time you should need to work on the assignments. If you don’t come to the lab, you are responsible for the assignment on your own, typically working on the weekend and Monday to Wednesday. The assignment is handed Wednesday in Brightspace with complete source code so that we can run your code to test it out.

Course Work: Weekly assignment must be handed in on time typically in the range 20 to 30 marks depending on the degree of difficulty. When portions of the assignment are incorrect or missing, assignments that follow will need the missing work provided in order for the follow-up assignment to be able to work (marks not given back when fixed after the due date). No solutions are given which is one reason why attending the help lab is useful. No tests and no exam.

Programming language used: **Smalltalk** (recommended; runs on Windows only) or **Swift** (not recommended because it is substantially more complex and more work; runs on a Mac (first choice) or Unix (second choice)).

Notes: Available through brightspace

Textbook: None although you may reference any compiler book using **LR(k) technology.** The course, however, deals with more advanced technology.

Course Outline

Novelty

Compilers consist of a 3 translators: scanner, parser, code generator. The details are NOT traditionally introduced in that order. Instead, we consider code generators first making it clear that translation is the essence of each stage.

Introduction:

A quick review of the components of a compiler, table driven scanners, table driven tree building parsers, tree walkers for code generation. A quick review of the process of designing a language for which a compiler is to be built.

Scanner/Parsers

A discussion of the details of a table driven scanner/parser and how it works.

The infrastructure for parsers/scanners

Detailed discussion of finite state machines (FSMs) and regular expressions (REs) and their use in transduction grammars. This includes operations such *,+,&,|,-, and . (concatenation).

Converting: A transduction grammar with regular expression right parts is converted into a transduction grammar with FSM right parts. This is a tree walking process akin to code generation but used instead in the context of grammar conversion.

Theoretical underpinnings for transduction grammars

Regular grammars, context free grammars, regular right part grammars, parse trees versus abstract syntax trees, handles, regular right part transductions grammars, LL(k) versus LR(k) grammars.

Constructing tables for regular right part grammars

More than shift-reduce tables; specifically, how to construct readahead FSMs for finding the right end of a handle, readback FSMs for finding the left end of a handle, and semantic action states for tree building along with their conversion into table formats for use by scanners/parsers. Overview of the process followed by detailed coverage.

Symbol tables

Techniques for dealing with scoping in typical programming languages.

Code Generation Basics

Machine versus virtual machine instruction sets. Code generation basics for virtual machine instruction sets. The distinction between expression contexts which requires a value on the virtual machine stack versus statement contexts which do not with several examples of language constructs that can be in used in both contexts.

More advanced code generation

Dealing with if statements, while loops, for loops. Dealing with short circuit boolean operations such as &&, ||, and ! which DO NOT require the use of And, Or, or Not virtual machine instructions.

Other Topics

Chain reduction optimizations for parsers. Non-canonical parsers.

Undergraduate Academic Advisor

The Undergraduate Advisor for the School of Computer Science is available in Room 5302C HP; by telephone at 520-2600, ext. 4364; or by email at undergraduate_advisor@scs.carleton.ca. The undergraduate advisor can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisor will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

SCS Computer Laboratory

SCS students can access one of the designated labs for your course. The lab schedule can be found at: https://carleton.ca/scs/tech-support/computer-laboratories/. All SCS computer lab and technical support information can be found at: https://carleton.ca/scs/technical-support/. Technical support is available in room HP5161 Monday to Friday from 9:00 until 17:00 or by emailing support@scs.carleton.ca.

Course Takeaway

A working compiler construction tool that can be used for building mini-languages in dozens of game programming applications.

Artificial Intelligence

With the advent of large language models (LLMs), it's becoming more and more possible to use chatbots such as ChatGPT, Copilot, Google Bard, Bing Chart, Cursor, research assistants such as Elicit, and image generators such as Stable Diffusion, Dall-E to aid with course work. However, there are problems with those tools:

- They sometimes hallucinate; i.e. lie and don't know it.
- They can't explain why something is true.
- They can't attribute knowledge (they don't know where they found this information).
- They can't learn and remember previous sessions (though there is some work trying to improve on this ascpect)
- They don't know if they don't know something; i.e., always give an answer whether right or wrong.
- They don't understand non-linguistic knowledge; e.g., liquids filling containers until full, batteries discharging until they don't work, 3D relationships between people and objects (e.g., right, left, above, below).
- They have no sense of ethics, danger, or reasonableness (e.g., a 1000 year old person).

This course was NOT designed to be completed in groups. Instead, the assignments must be done by individuals working alone but we do allow and in fact recommend LLMs as a programming aid provided the student is responsible for it's suggestions and provided the student acknowledges it's help; e.g., if it suggests implementing a specific function, add a comment such as "Aided by Copilot".

University Policies

For information about Carleton's academic year, including registration and withdrawal dates, see Carleton's Academic Calendar.

Pregnancy Obligation. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, visit Equity Services.

Religious Obligation. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, visit https://carleton.ca/equity/focus/discrimination-harassment/religious-spiritual-observances/.

Academic Accommodations for Students with Disabilities If you have a documented disability requiring academic accommodations in this course, please contact the Paul Menton Centre for Students with Disabilities (PMC) at 613-520-6608 or via Academic Accommodation or pmc@carleton.ca for a formal evaluation or contact your PMC coordinator to send your instructor your Letter of Accommodation at the beginning of the term. You must also contact the PMC no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). After requesting accommodation from PMC, meet with your instructor as soon as possible to ensure accommodation arrangements are made. For more details, visit the Paul Menton Centre website.

Survivors of Sexual Violence. As a community, Carleton University is committed to maintaining a positive learning, working and living environment where sexual violence will not be tolerated, and survivors are supported through academic accommodations as per Carleton's Sexual Violence Policy. For more information about the services available at the university and to obtain information about sexual violence and/or support, visit: Sexual Violence Prevention & Survivor Support | Equity and Inclusive Communities

Accommodation for Student Activities. Carleton University recognizes the substantial benefits, both to the individual student and for the university, that result from a student participating in activities beyond the classroom experience. Reasonable accommodation must be provided to students who compete or perform at the national or international level. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, see the policy.

Student Academic Integrity Policy. Every student should be familiar with the Carleton University student academic integrity policy. A student found in violation of academic integrity standards may be awarded penalties which range from a reprimand to receiving a grade of *F* in the course or even being expelled from the program or University. Examples of punishable offences include: plagiarism and unauthorized co-operation or collaboration. Information on this policy may be found here.

Plagiarism. As defined by Senate, "plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one's own". Such reported offences will be reviewed by the office of the Dean of Science. Standard penalty guidelines can be found here.

Unauthorized Co-operation or Collaboration. Senate policy states that "to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis". Please refer to the course outline statement or the instructor concerning this issue.