

# COMP 442 / 6421

# Compiler Design

## Tutorial 1

## Introduction

Instructor:

Dr. Joey Paquet

[paquet@cse.concordia.ca](mailto:paquet@cse.concordia.ca)

TAs:

Haotao Lai

[h\\_lai@encs.concordia.ca](mailto:h_lai@encs.concordia.ca)



# Lab Instructor

Section: lab hours NNJ M----- 15:45-17:30 H817  
lab hours NNI M----- 20:30-22:20 H817

Name: Haotao Lai (Eric)

Office: EV 8.241

Email: [h\\_lai@encs.concordia](mailto:h_lai@encs.concordia)

Website: <http://laihaotao.me/ta>



# Objective of the laboratory

1. help with the assignment / project
2. give hints for the implementation of the assignment / project
3. answer questions related to this course
4. there will be some tutorials during the lab (but not all the time)
5. please go the website to check if there is tutorial: <http://laihaotao.me/ta/comp442/w19.html>

# Introduction of the Assignment / Project

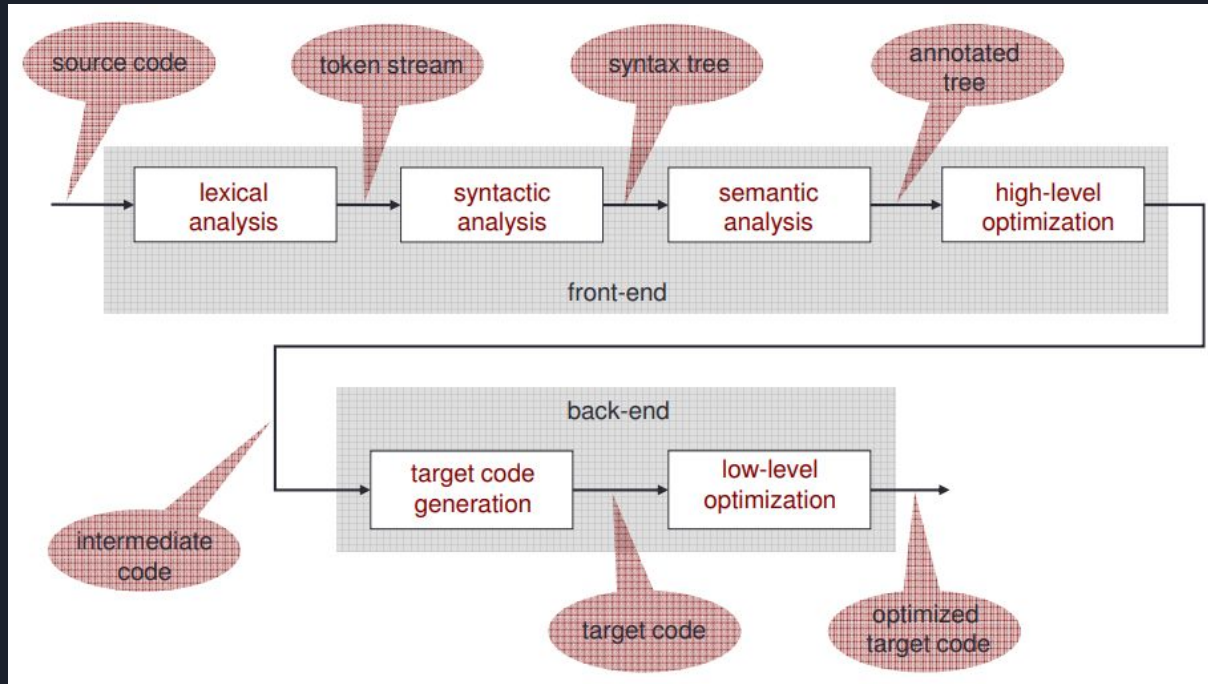
During this semester for the course compiler design, there are total 4 assignments you need to code. At the end of the term, you need to structure these 4 assignments to become a final project and demonstrate it in front of the course instructor. According to the course outline, each assignment is 10% of the final grade and the project is 30% of the final grade.

The goal of the project is to implement a compiler that can take source code as input and output the assembly code for a virtual machine (or simulator) called MOON.

```
1  class M {
2      int x;
3      int y;
4  };
5
6  program {
7      M m[3];
8      m[1].x = 64;
9      m[1].y = 1;
10     put (m[1].x + m[1].y);
11  };
```

```
1      entry % =====program entry=====
2      align % following instruction align
3      addi R1, R0, topaddr % initialize the stack pointer
4      addi R2, R0, topaddr % initialize the frame pointer
5      subi R1, R1, 24 % set the stack pointer to the top position of the stack
6      addi R14, R0, 64 %
7      sw -20(R2), R14 %
8      addi R8, R0, 1 %
9      sw -16(R2), R8 %
10     lw R6, -20(R2) %
11     lw R9, -16(R2) %
12     add R6, R6, R9 %
13     putc R6 %
14     hlt % =====end of program=====
```

# Introduction of the Assignment / Project





# Introduction of the Assignment / Project

| Assignment No. | Assignment topic   | Assignment Difficulty |
|----------------|--------------------|-----------------------|
| 1              | lexical analyser   | ★                     |
| 2              | syntactic analyser | ★★★                   |
| 3              | semantic analyser  | ★★                    |
| 4              | code generator     | ★★★★★                 |

*ps: keep in mind that it is not an easy assignment / course*

# Introduction of the Assignment / Project

## Project description


- Design and coding of a simple compiler
  - Individual work
  - Divided in four assignments
  - Final project is graded at the end of the semester, during a final demonstration
  - Testing is VERY important
- A complete compiler is a fairly complex and large program: from 10,000 to 1,000,000 lines of code.
- Programming one will force you to go over your limits.
- It uses many elements of the theoretical foundations of Computer Science.
- It will probably be the most complex program you have ever written.



# Introduction of the Assignment / Project

- When you implement your assignments, you can choose whatever programming language you want. But in the lab, I only support Java officially (and C++ personally).
- Start the assignment as soon as possible, there is no assignment you can finish within several hours in this course.
- You are not allowed to use any compiler auto-generation tool, such as but not limit to Lex, Yacc, LLVM, etc. In the other word, you must write your compiler from scratch.





# Submit your assignment

- Please follow exactly what the assignment description says, submit your documentation, code and any other required item via EAS before the due day.
- You need to provide the testing input you use to verify your program also the corresponding output you program generate. (the test case should be enough to cover most of the situation)
- Since you are allowed to use any language, you need to provide clear instruction and try your best to make the marker can run your code easily (like scripting the environment configuration process).

Q/A ?

