Syllabus

Fan Long

University of Toronto

UNIVERSITY OF TORONTO

Course Information

- Instructor: Fan Long (BA 3250)
- Contact Info: fanl@cs.toronto.edu
- Office Hours: Thursday after the class. Or schedule with email.
- Lectures: Thrusday 13:00-15:00 EST (MP 134)
- Tutorial: Tuesday 13:00-14:00 EST (MP 134 or zoom)
- References: Charles Fischer, Ron Cytron and Richard LeBlanc Jr. , Crafting a Compiler, Addison-Wesley 2009

  LLVM Infrastructure websites https://llvm.org

Course Information

- Marking: MarkUS, link TBD
- Web Page: https://q.utoronto.ca/courses/380291
- Bulletin Board: https://piazza.com/class/m5k5ycw6bvd553/
- Slides and Handouts:

  Will be posted in Quercus

Important Infos of CSC488

- Restructured course content to focus on **LLVM**
- The course project is based on **C++** rather than Java
- The project is designed for **individuals** rather than groups

- Tutorial format: TA talk about project logistics. If the tutorial is demo-heavy, it will be in zoom.
- No mid-term exam
- Open book final exam

Course Project

- Design and implement a small compiler for MiniC (a toy language)
- The compiler will be based on **LLVM** and therefore be written in **C++**
- Project has **7** phases/assignments
- Code templates will be given for each assignment except the last one
- Work **individually** and **independently** to finish the project
- Roughly **1k-2k** lines of code in total for all assignments
- Project contributes to **75%** of the final mark. **Start early!**

Cource Project Requirement

- A PC with Linux environment or a virtual machine that runs Ubuntu 20 or 22
  - On the first assignment, you will build/install ANTLR4, LLVM 15.0, and Clang 15.0 to setup your project environment.
  - Mac OS may work as well but it is not recommended.
  - Windows is **strongly not** recommended.

- **C++** skills are very useful. We will have tutorials to help on that.
- Because LLVM infrastructure is C++ based, it is almost impossible to use other programming languages. Our code template is also in C++.

Project Assignments & Marking

- Assignment 1 (5%) Prepare environment
- Assignment 2 (10%) Revise grammar and build parser
- Assignment 3 (11%) Build AST Tree
- Assignment 4 (12%) Symbol tables and semantic checking
- Assignment 5 (20%) LLVM IR generation
- Assignment 6 (11%) IR optimization
- Assignment 7 (6%) Optimization Competition
- Final Exam (25%)

Course Schedule

- Jan 9, First class
- Jan 22, Assignment 1 Due
- Jan 29, Assignment 2 Due
- Feb 12, Assignment 3 Due
- Feb 17, Reading Week, no class
- Feb 26, Assignment 4 Due
- March 18, Assignment 5 Due
- April 1, Assignment 6 Due
- April 4, Assignment 7 Due
- April 7-18, Final Exam

Course Content

- Introduction
- Parsing Techniques (Lexical and Syntax Analysis)
- AST Trees and Symbol Tables
- Semantic Analysis
- LLVM IR
- IR Code Generation
- Optimizations
- Runtime & Backend Code Generation

Course Project Submission Policies

- Everyone has a grace period of **96 hours** for late for the semester.
- For late beyond the grace period, **1%** penalty is applied per hour
- Sample solutions and test cases will be posted **4 days** after the submission deadline so **no late submission is allowed** after this point.
- If an exception is indeed required, we may approve to shift the mark of the missed submission to future assignments/exams. We will calculate your mark based on your average scores on other assignments.
- However, the **maximum** you can obain in this way is **75%** of the missed assignment. The only exception for this rule is student who add this course and request to shift weights for early assignments.
- You must complete at least **2 out of assignments 3-6** to receive score in this course.

Course Project Submission Policies

- A student may attempt a second submission within **10 days after the initial deadline** to fix bugs based on the released hidden cases. Fixed cases will allow the student to retain 75% of marks lost on the cases.
- The second submission must be modifications on the student own code base (not copying sample solutions) and contain descriptions on the root cause of the bugs.
- There is no second submission for assignment 1 and 7.
- The assignments are incremental, i.e., future assignments depend on previous ones.
- The student has the freedom to choose continue future assignments based on its own code base or the released sample code.

Course Project Submission Policies

- Discussion is encouraged, but plagirism is not tolerated.
- You are encouraged to share your thoughts and ideas, but not code.
- Offenders will receive zero on the corresponding assignment.

- **Please refrain from posting your code or sample code online**, even after the submission deadline, we may reuse the course project in future years.