COMP 442 /6421 Compiler Design

LAB 1:(Introduction)

Instructor:	Dr. Joey Paquet	paquet@cse.concordia.ca
TAs:	Hamed Jafarpour	hamed.jafarpour@concordia.ca
	Vashisht Marhwal	vmarhwal97@gmail.com

Outline

- Why Compiler Design?
- What the project entails
- Recommendation about the project
- Project - Language Choice
- Project – Testing
- Project - Version Control and Backups
- Recommendation of theoretical computer science review

Why Compiler Design?

- Compilers/Interpreters are a fundamental tool in programming
  - Making and customizing your own tools will make you a coding wizard.
  - Gain valuable insight into how compilers work and what limitations they have.
    - What can compilers do, and what *can't* they do?
  - Create simple yet powerful *Domain Specific Languages* to express ideas in a programming style, in non programming domains.

Why Compiler Design? - DSL

DSL: *Domain Specific Language*

- A specialized *"programming"* language which allows writing *"applications"* for a specific domain
- Contrast with General Purpose Language (C, C++, C#, Java, Python, etc.)
- Can be similar to a programming language, for a specific platform:
  - HTML
  - Unix Shell Scripts
  - SQL
- Or designed for a specific task:
  - Mathematics: *Maple*, *Wolfram*, *R*
  - Document editing: *LaTeX*, *Markdown*, *Emacs Lisp*
  - Software building: *Gradle*, *CMake*, *Maven*
  - Static analysis tools: Linters, Style checkers, Bug finders, etc.
  - Compute device programming: *GLSL*, *OpenCL C*
  - Music: *Csound*, *Sonic Pi*
  - Anything really: game level design, animation, drawing, chemistry, accounting, you name it!

What the project entails

- The project is about the design and implementation of a compiler for a simple programming language.
- The project is divided into five assignments.
- Each assignment corresponds to the design and implementation of a major component of the compiler, and makes use of the code base of all previous assignments.
- Thus, the project involves a very substantial amount of incremental coding.
- You can write the compiler in any language you are proficient with.
- You are allowed to use any computer that is available to you for the implementation.
- You are not allowed to use compiler-generation tools.

What the project entails

- The project is due on the last week of classes, where final project demonstrations are to be done individually with the instructor. No extensions of this deadline is possible.
- Students are encouraged to discuss the design and implementation issues of the project among them. However, each student must work on his/her individual implementation of the project.
- Note that you are responsible for the design of a complete set of tests for each part of the project.
- You are encouraged to cooperate with other students on this matter.
- Testing will be a major grading element of the assignments and the project.

Recommendation about the project

1. The project is the cumulative result of the 5 assignments
2. If you fall behind, catching up will be difficult
3. Starting early.

Project - Language Choice

You can use any language

- Pick a language you're familiar with
  - Now is not the time to learn a new language
- Java is supported in the labs and by the TAs

Recommendation: Pick a language where the following is easy . . .

Project - Testing

Manual testing

- Consistency in tests, inputs and results
  - Test files

Automatic testing

- Important for validating your compiler
- Compilers are straightforward to test, since they are stateless at a high level
  - Given an input, they produce an output (source -> tokens)
- Test cases can be made easily from the assignment specifications
- Test often!
- Your tests should be *easy* and *fast* to run

Project - Version Control and Backups

Version control

- Important for any software project
- Very important for a complex software project which are prone to errors, i.e. compilers
- If you haven't done so before, now is a good time to start using version control
  - *SCS Concordia* frequently offers tutorials on the version control system ***Git***

Backups

- **Please, please, please backup your assignments**
  - If using version control, repository systems (GitHub, BitBucket)
    - Free for students
  - Dropbox, OneDrive, email, external hard drive, USB stick
    - ***ANY*** backup is better than none
- Make sure your backups are private, and accessible only to you
  - Not doing so constitutes an **academic offense** under the *Academic Code of Conduct*
  - Private repositories

Recommendation of theoretical computer science review

The following topics are the foundation to this class, and compiler design in general. Reviewing them is recommended

- Regular Languages
  - Finite State Automata
  - Regular Expression
  - Conversion between the two
- Context Free Grammars
  - Derivation process
  - Push-down automata

Thanks!