Fun with compilers: A practical introduction to compilers and LLVM

1. Introduction to compilers
2. Phases of a Compiler
3. Practical introduction to compiler frontend
    a. Lexical analysis (lex)
    b. Syntax analysis (yaac)
    c. Simple lex and yacc tutorial
    d. Building a calculator with lex and yacc
    e. Building a toy language
4. Intermediate language
    a. DAG
    b. 3 address code
    c. Basic blocks
    d. Flow graphs
5. Introduction to LLVM and LLVM-IR
6. Introduction to optimizations
    a. Basic optimizations (Dead code elimination, strength reduction, etc.)
    b. Basic loop optimizations (Fission, Fusion, etc)
7. Dependence analysis
8. Overview of compiler backend and backend optimizations
9. Data flow analysis
    a. Reaching definitions
    b. Live variables
    c. Code motion
    d. Dead code elimination using LLVM
    e. Constant propagation using LLVM
10. Optimizing Loop codes
    a. Dominators
    b. Back edges
    c. Loop representation in LLVM
    d. Loop invariant code motion in LLVM
11. Advanced dependence analysis and loop transformations
    a. Dependence representations
    b. Advanced loop transformations (Tiling, Skewing, etc.)
    c. Transformation legality

Course Materials:

- Text book: Compilers: Principles, Techniques, and Tools -- Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey Ullman
- Others: Additional materials, if any, will be provided by the instructor

Pre req:

- Hard
  - 260 – Introduction to Computer Architecture
- Soft *
  - 355 Programming Language Design
  - 317 Automata and Formal Languages
  - 360 Systems Programming C/C++
- Language
  - C++ *

* If these requirements are not satisfied, special permission from the course instructor is required to enroll in this course