CSE 504 Compiler Design
Lecture 1

R. Sekar

Course Objectives

- **Gain an in-depth understanding of how compilers are designed and implemented**
  - Underlying theory
  - Hands-on work
  - Experience first-hand how theory and practice come together
- **Give you the background to develop**
  - Full-scale compilers and interpreters for traditional or scripting languages
  - Parsing complex input and configuration files
- **Secondary objective:**
  - Learn to develop moderate-size programs in C++11/14
    - Object oriented programming
    - Using Standard Template Classes (STL)
    - Development tools: compilers, debuggers, make, …

Prerequisites

- **CSE 307 or equivalent**
  - Programming language concepts
    - not programming with different languages
  - Please review the lecture notes for CSE 307 to refresh yourself on this topic:
    - http://seclab.cs.sunysb.edu/sekar/cse307/
- **CSE 304 or equivalent**
  - Undergraduate compilers course
  - I won't enforce this

Course Organization

- **Regular Languages, Lexical analysis (3)**
- **Context-Free Grammars and Parsing (4)**
- **Attribute Grammars (2)**
- **Types and type-checking (2)**
- **Runtime Environments (4)**
  - Memory allocation, parameter passing, …
- **Code generation (4)**
- **Optimization and other Advanced Topics (7)**

Grading

- See course web page

Academic Dishonesty

- **Any form of copying in programming assignments will lead to an "F" grade in the course**
  - Copying from friends or the Internet
  - Working in groups
- **Use CSE 504 discussion forum for discussions**
  - Any discussion on assignments among friends is appropriate ***only if*** you'd have the same discussion in my presence
- **We use various measures to discourage cheating**
  - Although the assignments build on each other, you don't have to rely on your code from previous assignment for the next one --- we will provide you good, working code
  - There will be an oral interview at some point in November when you will be asked to explain various points about your code
  - We use software to detect copying. This software can detect copying of small snippets, and can easily deal with common techniques for evasion such as variable renaming and many more that most people don't even think about

Survival Tips

- **Make sure that you have a reasonable knowledge of C++, or make sure that you get up to speed very quickly**
- **Don't postpone working on assignments**
- ***Start early on assignments!***
- ***Design before you code, follow good programming practices***
  - The best-written programs are well thought through, compact, and well-commented
  - If you don't do this, your programs will become totally unmanageable after a while
- **Follow the discussion forum**