

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++
 - ▼ 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
- ◆ **CSE 304 or equivalent**
 - Undergraduate compilers course
- ◆ **I will insist on *one of* these prereqs, unless there are exceptional circumstances**
 - To brush up on the concepts, you will be given access to CSE 307 lectures/slides

Course Organization

- ◆ **Regular Languages, Lexical analysis (4)**
- ◆ **Context-Free Grammars and Parsing (4)**
- ◆ **Attribute Grammars (3)**
 - Semantic analysis (type-checking etc.)
- ◆ **Runtime Environments (4)**
 - Memory allocation, parameter passing, ...
- ◆ **Interpreters (2)**
- ◆ **Code generation (2)**
- ◆ **Optimization and other Advanced Topics (8)**

Grading

- ◆ **Exams (50% to 55%)**
 - One or two midterms
 - Final
- ◆ **Programming assignments (40%-45%)**
 - Symbol Table: 5% Due: 2/6
 - Lexical analysis: 7% Due: 2/20
 - Parsing: 9% Due: 3/6
 - Abstract Syntax Tree 11% Due: 3/20
 - Type-checking 11% Due: 4/3
 - Interpreter 10% Due: 5/1
(Extra credit)
- ◆ **Attendance, class participation: up to 5%**



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 during April 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++, and tools such as debuggers. Otherwise, don't take this course**
- ◆ **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

