CS243
Course Staff
Administrivia
In Praise of Compiler Technology
Course Staff

◆ Faculty:
  ✤ Wei Li (Intel Corp.)
  ✤ Jeff Ullman

◆ TA’s:
  ✤ Ben Livshits
  ✤ Hasan Imam

◆ Info: http://cs243.stanford.edu
◆ Email: cs243staff@gmail.com
Course Requirements

- Midterm (25%) and Final (45%).
- Programming project (teams of 2 OK; 20%).
- Gradiance on-line homework (10%).
Textbook

◦ On-line access to new Dragon-book chapters + Gradiance service at www.aw-bc.com/dragonbook

◦ After signing up, join the CS243 class at www.gradiance.com/pearson by entering class code 659152EA.
A Word About Gradiance

◆ It looks like multiple-choice, but it isn’t.
◆ You really have to solve the problems, and the system then samples your knowledge.
◆ If you err, you get a hint and place to read, and you are allowed to try again.
Why Study Compilers

1. Excellent software-engineering example --- theory meets practice.
2. Essential software tool.
3. Influences hardware design, e.g., RISC, VLIW.
4. Tools (mostly “optimization”) for enhancing software reliability and security.
Modern architectures have very complex structures, especially opportunities for parallel execution.

Sequential programs can only make effective use of these features via an optimizing compiler.

**Hardware question**: If we implemented this, could a compiler use it?
Software Reliability

◆ Optimization technology (data-flow analysis) used in:
  - Lock/unlock errors.
  - Buffers not range-checked.
  - Memory Leaks.
  - SQL injection bugs.

◆ Ben will talk about these.
What CS243 Offers

- Compiler methodology for both compiler implementation and related applications.
- Theoretical framework.
- Key algorithms.
- Hands-on experience.
- **Nongoal**: build a complete optimizing compiler.