



# *Introduction & Overview*

*CS4533*

*from Cooper & Torczon*

## Syllabus

---



- Overview
- Lexical Analysis (Scanning)
- Syntactic Analysis (Parsing)
- Context-Sensitive (Semantic) Analysis
- Intermediate Representations
- Symbol Tables
- Instruction Selection
- Register Allocation
- Instruction Scheduling

## Compilers

---



- What is a compiler?
  - > A program that translates an *executable* program in one language into an *executable* program in another language
  - > The compiler should improve the program, *in some way*
- What is an interpreter? A program that reads an *executable* program and produces the results of executing that program
- C is typically compiled
- Scheme is typically interpreted
- Java is compiled to bytecodes, which are then interpreted

This course deals mainly with *compilers*

Many of the same issues arise with *interpreters*

*Classic qualifying examination question*



## Why study compilation?

---

- Success Stories
  - > Application of theory to practice
    - Scanning, parsing, static analysis, instruction selection
  - > Many practical applications have embedded languages
    - Commands, macros, formatting tags ...
- Humbling failures
  - > Problems that are truly hard
    - Can't check automatically if a grammar describes the language
    - Can't check automatically if a grammar is ambiguous
  - > Requires *ad hoc* handling of some issues
- Practical algorithmic & engineering issues
  - > Approximating hard problems
  - > Emphasis on efficiency & scalability
  - > Small issues can become important *(as in real life)*

## *Intrinsic interest*

---



- > Compiler construction involves ideas from many different parts of computer science

<i>Artificial intelligence</i>	Greedy algorithms Heuristic search techniques
<i>Algorithms</i>	Graph algorithms, union-find Dynamic programming
<i>Theory</i>	DFAs & PDAs, pattern matching Fixed-point algorithms
<i>Systems</i>	Allocation & naming, Synchronization, locality
<i>Architecture</i>	Pipeline & hierarchy management Instruction set use

## *Experience*

---



- You have used several compilers.
- What qualities do you want in a compiler that you buy ?



## *Experience*

---

- You have used several compilers.
- What qualities do you want in a compiler that you buy ?

1. Correct Code
2. Output runs fast
3. Compiler runs fast
4. Compile time proportional to program size
5. Support for separate compilation
6. Good diagnostics for syntax errors
7. Works well with debugger
8. Good diagnostics for flow anomalies
9. Good diagnostics for storage leaks
10. Consistent, predictable optimization