Topics: overall organization

- Overall Compiler Organization
  - what's in the front/back end
  - what the job of each phase is

Topics: scanning

- Terminology: lexeme, token, pattern
- Regular expressions
  - the notation
  - how you build them, concrete examples
  - advantages and limitations
- Scanners
  - 4 steps: RE -> NFA -> DFA -> code/tables
  - constructing an NFA
  - constructing a DFA

Topics: parsing

- Grammars
  - BNF
  - how you build them, concrete examples
  - Advantages & limitations
- Derivations
  - abstract vs. concrete (parse) syntax trees
- Parsing algorithms
  - top-down vs. bottom-up
  - predictive parsing
  - implementation

- LL(1) grammars
  - ambiguity & fixes
  - meta-rules, grammar rewrites, language changes
  - Common prefixes
  - left recursion
- Bottom Up LR(0) Parsing
  - Building a LR(0) DFA
  - Building a LR(0) parse table
  - Shift-reduce parsing using an LR(0) parse table

Topics: semantic analysis

- Type checking basics
  - examples
  - type checking algorithm
- Symbol table
  - attributes for each type of symbol
  - scopes: when created, how used
  - handling procedures
- Strong, weak, static, dynamic typing
- Type equivalence
  - structural vs name

The midterm

- 50 minutes, open book/open note, no open computing devices
- Kinds of questions may include
  - T/F explain
  - Short answer
    - "The following language features are from real languages. Explain which of the three phases – lexing, parsing, semantic analysis/typechecking – is most directly affected by these features and why."
  - Others
    - Homework-like questions
    - Other