HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

CS153: Compilers
Lecture 26:
The Economics of Programming Languages
Guest Lecturer: Evan Czaplicki

Stephen Chong

https://www.seas.harvard.edu/courses/cs153

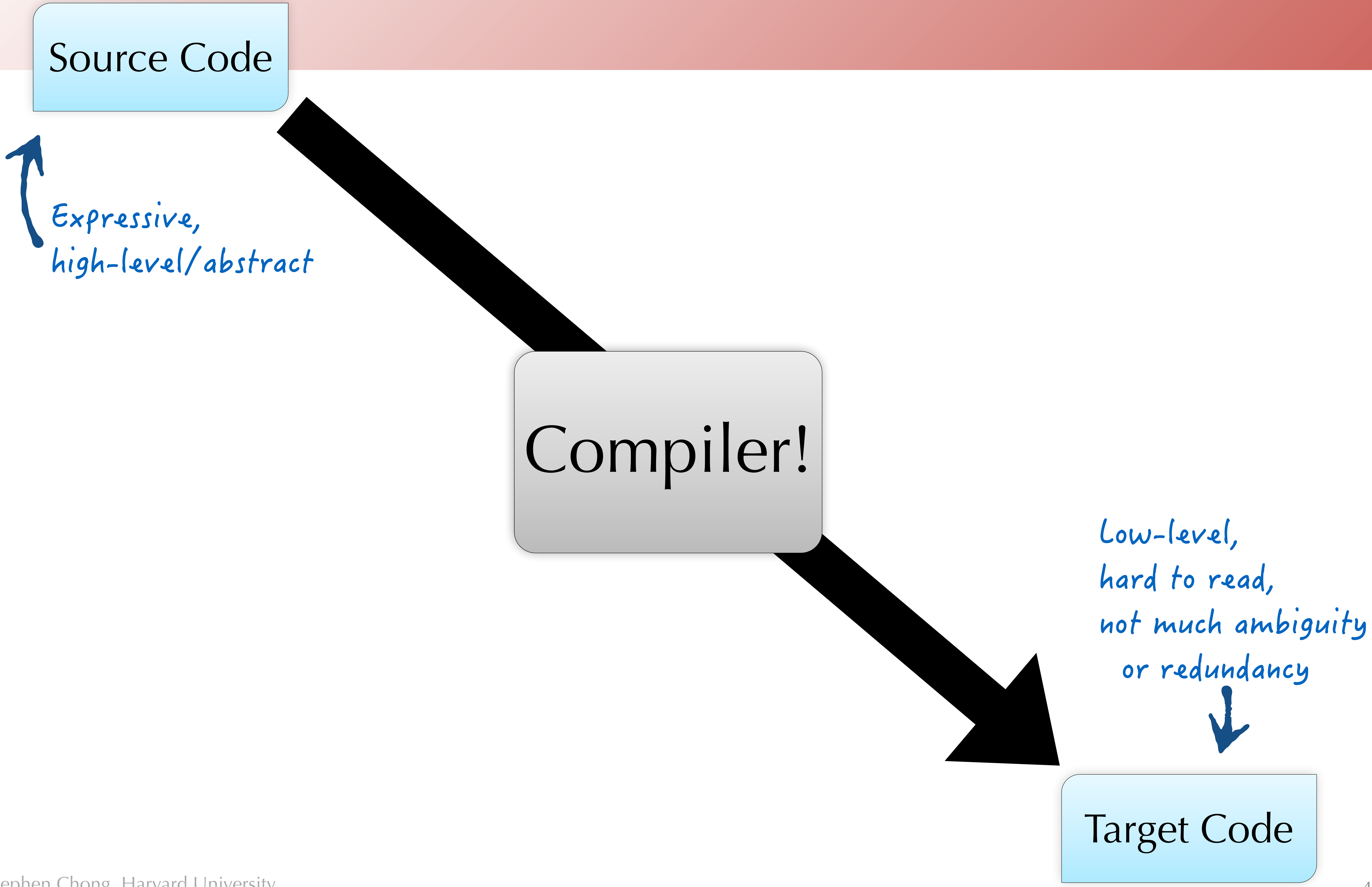
Announcements

- HW6: Optimization and Data Analysis
  - Due today (Tue Dec 3)

The Economics of Programming Languages

- Evan Czaplicki ’12
  - Creator of the Elm programming language
  - https://elm-lang.org/

What is this course about?



Source Code

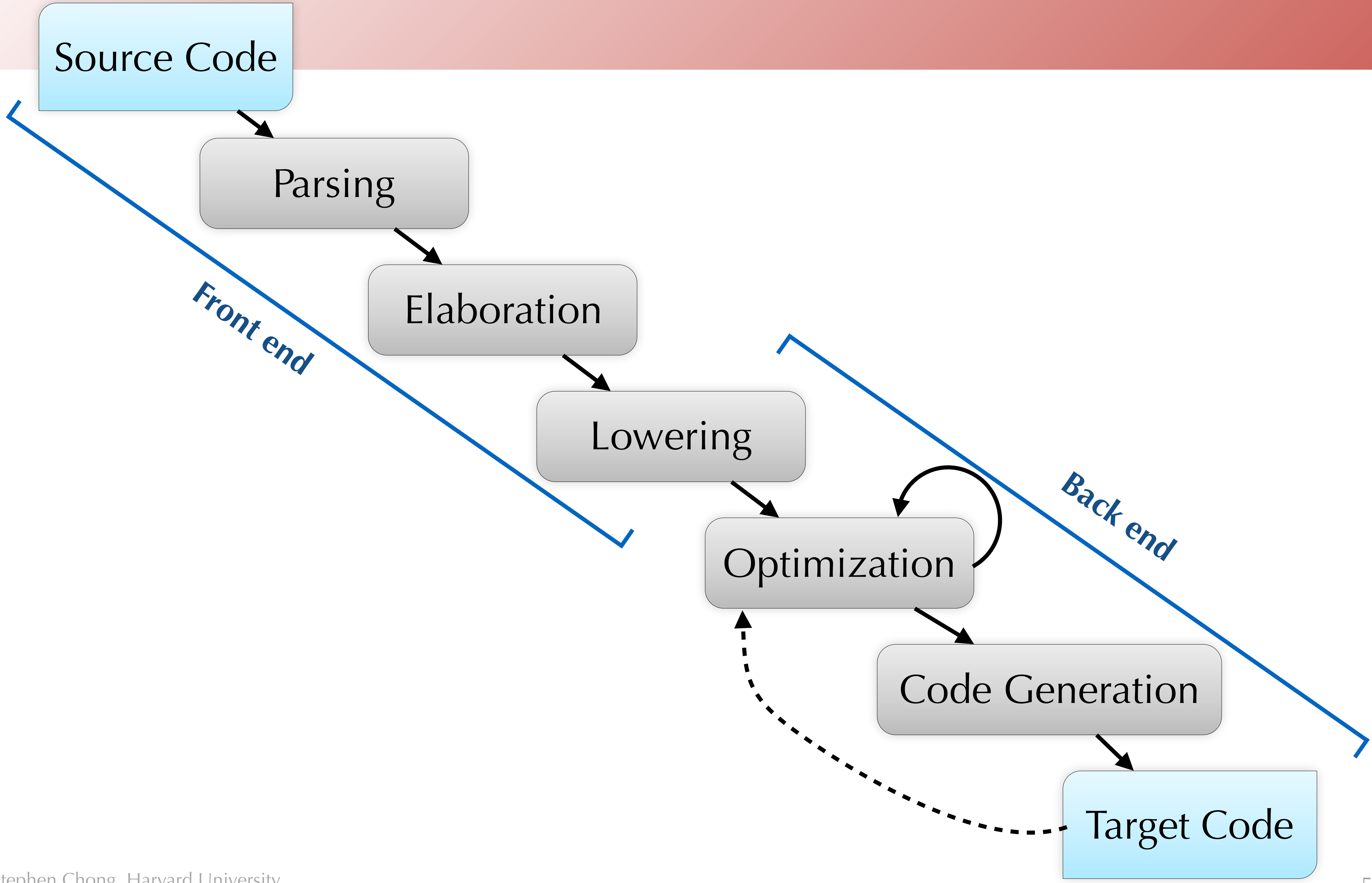
Expressive,
high-level/abstract

Compiler!

Low-level,
hard to read,
not much ambiguity
or redundancy

Target Code

Basic Architecture



Topics

- Lectures 2 + 3: Assembly
  - Turning C into machine code
  - Intel x86
  - x86lite
  - C memory layout
  - Calling convention
- Lecture 4,5,6: Intermediate Representation
  - Compiling expressions directly to assembly
  - Motivating Intermediate Representations (IRs)
  - Simple Let Language
  - Basic blocks
  - Control-flow graphs
- Lecture 7: LLVM, Structured Data in LLVM
  - Arrays
  - Tagged datatypes (and switches)
  - Datatypes in LLVM
- Lecture 8: Lexing
  - Tokens
  - Regular Expressions
  - Deterministic Finite Automata
  - Nondeterministic Finite Automata
  - NFA to DFA
  - Lexer Generator
- Lecture 9: Recursive Parsing
  - Context-free grammars
  - Derivations
  - Parse trees
  - Ambiguous grammars
  - Recursive descent parsing
  - Parser combinators
- Lecture 10: LL Parsing
  - Nullable, First, Follow sets
  - Constructing an LL parsing table
- Lecture 11: LR Parsing
  - Constructing a DFA and LR parsing table
  - Using Menhir
- Lecture 12: First-class Functions
  - Nested functions
  - Substitution semantics
  - Environment semantics and closures
- Lecture 13: Compiling Functions
  - Closure conversion
  - Implementing environments and variables
  - DeBruijn indices
  - Nested environments vs flat environments

Topics

- Lecture 14: Type Checking
  - Judgments and inference rules
- Lecture 15, 16: Subtyping
  - Types as sets of values
  - Subtyping
  - Subsumption
  - Downcasting
  - Functions
  - Records
  - References
- Lecture 17, 18: Compiling Objects
  - What is object oriented programming
  - Dynamic dispatch
  - Code generation for methods and method calls
  - Fields
  - Creating objects
  - Extensions
  - Type system
- Lecture 19: Optimizations
  - Safety
  - Constant folding
  - Algebraic simplification
  - Strength reduction
  - Constant propagation
  - Copy propagation
  - Dead code elimination
  - Inlining and specialization
  - Recursive function inlining
  - Tail call elimination
  - Common subexpression elimination
- Lecture 20: Dataflow Analysis
  - Liveness analysis
  - Worklist algorithm
  - Generalizing dataflow analysis
  - Available expressions
  - Reaching definitions

Topics

- Lecture 21, 22: Register allocation
  - Graph coloring by simplification
  - Coalescing
  - Coloring with coalescing
  - Pre-colored nodes to handle callee-save, caller-save, and special purpose registers
- Lecture 23: Loop Optimizations
  - Examples
  - Identifying loops
  - Dominators
  - Loop-invariant removal
  - Induction variable reduction
  - Loop fusion
  - Loop fission
  - Loop unrolling
  - Loop interchange
  - Loop peeling
  - Loop tiling
  - Loop parallelization
- Lecture 24: Embedded EthiCS module
  - Ethics of Open Source
  - Free/Open Source Software
  - Short History
  - Argument from Freedom
  - Economic Arguments
  - Identifying Possible Ethical Concerns Philosophical Tools: Compensation of Maintainers
- Lecture 25: Garbage Collection
  - Key idea
  - Mark and sweep
  - Stop and copy
  - Generational collection
  - Reference counting
  - Incremental collection, concurrent collection
  - Boehm collector

What Next?

- Exam
  - Will release some practice questions later this week
  - Will arrange review session, likely Monday Dec 16
- Other courses
  - CS152: Programming Languages
    - Spring. Prof Nada Amin
  - CS252r: Advanced Topics in PL
    - Spring 2020: Building a Verified Compiler
- Research
  - Come and chat!