

# CS 432 Final Study Guide

Fall 2020

In addition to all of the items listed on the midterm study guide, you are expected to be able to perform the following tasks on the exam:

- Discuss various architecture and runtime system considerations relevant to compilers.
  - Instructions, opcodes, operands, and assembly/machine code.
  - Standard linkage conventions and how the stack is used to enable procedure calls.
  - Heap allocation and deallocation strategies.
- Convert Decaf code into equivalent ILOC in static single-assignment form with the calling conventions discussed in class
- Split an ILOC function into basic blocks and draw the control-flow graph (CFG).
- Calculate data-flow analyses on an ILOC or three-address-code CFG:
  - Dominance and post-dominance
  - Liveness (variables or registers)
- Define *static single-assignment* (SSA) form and identify whether a given ILOC or three-address-code basic block is in SSA form.
- Name and define the various optimization scopes.
- Describe various optimizations and apply them to small code examples.
  - Local value numbering
  - Tree-height balancing
  - Superlocal value numbering
  - Loop unrolling
  - Constant propagation
  - Dead code elimination
  - Global code placement
  - Lazy code motion
  - Inline substitution
  - Procedure placement
- Compare and contrast local top-down, local bottom-up, and global graph-coloring-based register allocation.

- Perform bottom-up register allocation on an ILOC basic block.
- Perform list scheduling on an ILOC basic block.
- Define and discuss specific terms or vocabulary related to any of the above concepts, including a detailed description of why and how they are relevant to the construction of a compiler.
- Review the semester-long project and discuss how various Decaf design decisions affected the amount and complexity of the work done. Where the decisions simplified development, name concrete ways that the implementation would be more difficult if the decision was changed.