

# CISC 471/672 Study Guide for Second Exam

## Fall 2015 In-class Individual Exam

### 1 References

- Class-time notes and slides from October 13, 2015 through December 3, 2015.
- Assignments (PA3, PA4, PA5, LR(1) homework)
- Handouts from class - see the course calendar
- Quizzes
- For further reading, look at the textbook pages cited in the course calendar/syllabus if you need more details

### 2 Topic Coverage

1. Semantic analysis – kinds of checking at semantic analysis time and how implemented
2. Static and dynamic scoping rules
3. Implementing static scoping (symbol tables) – including block-structured (procedural) and OO languages
4. AST traversal to build and access a symbol table
5. Generation of code – function calls/returns, control structures, etc
6. Code generation for OO languages – vtables, object layouts, new object storage, object accesses, polymorphic call site handling
7. Activation record/frame layout and manipulation on the runtime stack
8. Run-time storage management – static, stack, heap, including garbage collection techniques (reference counting, mark-and-sweep, stop-and-copy, generational)
9. Register allocation through live variable analysis, interference graph construction and graph coloring

### 3 Format of Exam

The exam is closed book, closed neighbor and you will have the full final exam time period to work. In general, the exam will be a combination of testing your basic knowledge and understanding of the concepts covered in class and application of the concepts. Some example types of questions:

Partial credit will be given when possible on any question in the exam.

- True and false with justification
- Show scopes of variables
- Show symbol tables for a program
- Draw diagrams to show concepts of code generation and run-time storage management.
- Draw pictures of memory at different points during run-time.
- Make and justify compiler design decisions.
- Describe and justify which items can be stored on the stack, heap, static store.
- Short-answer questions
- Convert program segments into AVR code, such as polymorphic call sites, regular method call sites, method prologues and epilogues
- Display an AR frame, stack and heap elements
- Show the heap memory at certain points during garbage collection
- Show register interference graph for a given code segment
- Determine whether register interference graph can be colored with specified number of registers

### 4 How to Study

Review your lecture notes, handouts, labs, and textbook chapters. Concentrate on your lecture notes and handouts.