CISC 672: ADVANCED COMPILER CONSTRUCTION
Spring 2005
Second Exam Study Guide
Tuesday, May 17, 2005, class time, regular classroom

1 References

- Lectures notes from just after first exam through last day of lecture
- Textbook: See readings on schedule online for class lectures covered
- Programming assignments : PA4-semantic analyzer, PA5-code generator
- Handouts from class lectures on type inference and checking
- Code generation handouts

2 Topic Coverage

- inheritance graph construction
- type checking and type inference
- type checking rules through formal specification
- overloading, coercion, polymorphism and handling during type checking
- intermedate code representations
- simple code generation
- run-time storage management: static, stack, heap
- activation record layout and management
- generating code for function calls/returns
- generating code for nonlocal variables and parameters
- code generation for object-oriented languages
- heap management through garbage collection

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 to expect:
- short answer.
- draw diagrams to show concepts of code generation and run-time storage management.
- read and explain type rules.
- draw pictures of memory at different points during run-time.
- make and justify compiler design decisions.
- draw the representation of a particular data structure in the activation record.
- describe and justify which items can be stored on the stack, heap, static store.
- short essay.
- true/false or justify why something is true/false.

The questions are NOT multiple choice. Instead, partial credit will be given when possible on any question in the exam.

4 How to Study

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