CSE443
Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

http://www.cse.buffalo.edu/faculty/alphonce/SP17/CSE443/index.php
https://piazza.com/class/iybn4ndqa1s3ei
Final Exam

5/19/2017, Friday
8:00AM - 11:00AM
Talbert 107
Project Due Date

Wednesday
5/17/2017
5:00 PM
Special Just-For-You Office Hours

Monday 05/15/2017

9:00 AM to noon
2:00 PM - 5:00 PM
Exam format

- Expect about 4 short essay questions. We will use BlueBooks.
- You will have 3 hours to write the exam.
- I would expect you to take about 30 minutes per question, leaving about an hour to check over your work.
Possible Exam Questions

- Anything from HW1

... or ...
Type checking (Semantic processing)

Explain how type errors are detected. Discuss how type information is gathered, stored and checked. Pick a concrete syntactic construct that can contain a type error, and explain how type checking detects the error.
Intermediate Code Generation

Explain how short-circuit Boolean expressions are translated into intermediate code. Discuss how jump targets can be determined. Illustrate by showing how a concrete Boolean expression involving at least two Boolean operators is translated into intermediate code.
Register Allocation and Assignment

Describe the `getReg(I)` algorithm, answering the questions of what data structures it uses, when and how these structures are updated. What is meant by "spill", when does it occur, and why is it needed?
Symbol Table Usage

Describe the structure and use of a symbol table. Explain which phases of the compiler use the table, including what data is written to or read from the table during each phase.
Invocation Records

Describe a typical layout for an invocation record, detailing what information is stored in the record. Explain how variable length parameters and variable length local data can be accommodated. Discuss the location and use of the stack and top pointers.
**Function Calls**

- Explain how a function call takes place. Include in your discussion mention of the roles of the caller and callee in setting up the invocation record, and how machine state is remembered at the call and restored at return. Explain how recursive calls are handled (do NOT discuss tail-call optimization).
Optimizations

Pick an optimization and explain the benefit(s) of having the compiler apply it to code, and sketch how it works.

Ex:
- tail-call optimization
- code motion
- dead code elimination
Final exam questions?

Project questions?
Thanks for a great semester!
Congrats to everyone graduating!!
Congrats to everyone graduating!!

You are coming to the May 19th dinner, right?
Congrats to everyone graduating!!

And the May 20th ceremony?
Check out my cool hat!

There will be people in funny outfits!
Have a wonderful summer!
(and see you at the final)