CS 363 Review for Final. Please also look at Reviews 1 and 2 from earlier in the course.

1. What is a programming language? What should it contain?

2. Does a programming language describe a finite or infinite set?

3. What is the tradeoff between bottom-up parsing and the CYK algorithm?

4. What factors should you take into account when deciding which programming language to use?

5. What is the difference between an interpreted and a compiled language? What steps would you need to accomplish to create a new compiled language?

6. Name some programming language features that help the programmer write more reliable and readable code.

7. In a language that supports default parameters, how does one tell which parameters are specified and which are default. For example if a function is defined to take 4 parameters and only 2 have been specified in the call, how does the compiler typically decide which 2 these are?

8. Suppose P is a procedure. What does P's dynamic link point to? Assuming that the compiler supports static chaining, what does P's static link point to? Give an example (skeleton diagram) of a program in which the static links between functions are not the same as the dynamic links of how the functions are called.

9. What is the main benefit of using template functions in a language like C++ or Ada?

10. When using co-routines, what is the difference between calling a function and transferring to a function?

11. Write a Scheme function that returns the last item in a list.

12. Write a Scheme function to determine if a list is a palindrome. That is, if the first item is the same as the last, the second equals the second to last, etc.

13. During bottom-up parsing, why is it a good idea to maintain 2 stacks at the same time? What information is contained on each?

14. Draw the function instance graph for the file abcd.C on the class Web site. Why is this information useful?

15. Illustrate 2 ways in which multiple variables can share the same register. (Hint: one way deals with time, the other with space.)

16. What is the typical memory layout of a process? You need to show activation records, and distinguish between static, stack and heap. Where would we put static and dynamic links?

17. Among static data, run-time stack, and heap, which area should be the largest? Why?

18. What is the difference between a token and a lexeme?

19. During compilation, variables are kept in a symbol table. Which phase of compilation is most concerned with creating and checking the symbol table? What information about a variable is kept in it?

20. What is a basic block? Why does the loop detection algorithm refer to basic blocks instead of individual instructions?

21. What is a back edge? In the loop detection algorithm, how do we know if a back edge exists in a function?

22. Suppose you are the number 12 in the statement "x := y + 12;" in some compiled high-level language. Describe what happens to you during the process of compilation. Where is your final resting place?

23. In the compiler, at what point do we decide on the precedence and associativity of operators? How do we specify?

24. Why could traditional Fortran not support recursion?

25. What is the difference between syntax and semantics? What is the difference between a program's syntax tree and the parse tree?

26. What is meant by the "front end" of the compiler? What does it do?

27. What is intermediate code? Which optimizations can be performed on it? Which cannot?

28. When optimizing code, why is it a bad idea for the compiler to rearrange the order of the mathematical operators in an expression to exploit the associative and commutative properties of arithmetic? Give an example.

29. Consider the following grammar. Assume that nonterminals are enclosed in angle brackets. Draw a parse tree for the sentence: walk east pick box drop ball stop.
    <robot> → walk <dir> <robot> | pick <obj> <robot> | drop <obj> <robot> | stop
    <dir> → north | east | south | west
    <obj> → box | ball

30. In a top-down parsing table:
    a. What do the rows and columns correspond to?
    b. What is the significance of having an entirely blank column in the table?