CS 363 – Review for test 2

1. In what ways did the first Fortran compiler differ from compilers made today?

2. What semantic actions would we expect to perform when creating the syntax tree for this statement of variable declarations? Show how the syntax tree is therefore constructed step-by-step: `int a, b, c;`

3. The meaning of a computer program is determined not only by the list of instructions, but also on the order in which they are performed. This is control flow. What are the various ways in which we can specify the order of statement execution?

4. Depending on the context, the word "equal" has various meanings in a computer program. What definitions can you come up with?

5. How does the Fortran computed goto statement work? Here is an example:
   goto(10, 20, 30, 40), i

6. What is a packed record or packed array in Pascal? What is its main advantage and disadvantage?

7. Algol's for loop has been criticized for being baroque. What does this mean? Illustrate with an example.

8. What is an iterator?

9. How can you tell if a language exhibits strong typing or weak typing?

10. What is an enumeration type? How does a programming language represent this data type internally?

11. Is it possible for a programming language to be typeless? If so, how could it work?

12. Give two reasons why a programmer would want to use a subrange type.

13. Suppose an array, stored in column-major order, has 3 books, 3 pages, 3 rows, and 3 columns. Show which elements occupy the first 10 cells in memory – in other words, which cells occupy the 10 lowest addresses.

14. In Java, a Boolean expression used for an if() expression must be of boolean type, but in C/C++ it may be any integer type. What are the tradeoffs?

15. What is the disadvantage to having a small number of precedence levels in a language, such as in Pascal?

16. Suppose you were working with a language in which the and/or boolean connectives (analogous to && and ||) did not support short-circuit evaluation. Show how you would compensate.

17. Typical rules for case/switch statements are that the case values be constant, unique and of integral type. Explain why we have these rules.

18. Consider the switch statement in Java (similar to that of C and C++ as well). Suggest some improvements to its syntax, motivated by what you may have found in other languages.

19. Illustrate the "dangling else" problem. How is it usually resolved?

20. What are the two ways to determine if two types are "the same"? Which method is easier for the compiler?

21. Suppose `a` is a three dimensional array in the C (or C++) language. Write pointer notation equivalent to `a[0][3][0].`

22. Give an example in which a variant record would be an efficient use of space.

23. In an array of doubles, each element occupies 8 bytes. Suppose we define an array to contain 8 rows and 10 columns. If the address of element [5][4] is 1000, what is the base address of the array? Note there are 2 possible answers depending on whether the organization of the array is row major or column major.

24. If a 5 dimensional array has the following dimensions: 10x10x10x10x10, each element occupies 1 byte and the base address is 0, then what are the row-major and column major addresses of the element [4][7][8][1][2] (assuming the index numbering begins at zero.)

25. Explain what is meant by a "reference" type.

26. Which data type is usually implemented as a bit vector?

27. Why are there two possible versions of "call by value-result"?

28. Describe an experiment where you would determine if a parameter is passed by value or by reference.

29. Suppose a function has 10 blocks. The successor of each block is given below. For each block, determine its predecessors and dominators. Are there any loops? If so, for each loop, identify its header block, back edge(s), and which blocks comprise the loop.

Block	1	2	3	4	5	6	7	8	9	10
Successors	9	7	4	5	6	7	3, 8	9	2, 10	-