1. What are some of the advantages of using BNF to describe a programming language’s syntax?

2. Lexical analysis is usually separated from syntax analysis in a compiler. Why? What’s the difference between them?

3. How are reserved words handled by a lexical analyzer?

4. What’s the difference between a lexeme and a token?

5. What are the two categories of parsers? Describe each.

6. What are the advantages of an LR parser?

7. What are some concerns related to rules for naming identifiers?

8. What is the l-value of a variable? The r-value?

9. What is a binding?

10. What are the possible binding times for a programming language element?

11. What are concerns related to elementary data types?

12. What are the six attributes of a variable? Briefly describe each.

13. What is the purpose of a declaration? Why require them? Can’t the type of a variable be determined by its use? If so, why not just do that?

14. What is strong typing?

15. Distinguish between static and dynamic binding. What are the relative advantages and disadvantages? Give an example of each.

16. What are the relative advantages/disadvantages of explicit vs. implicit variable declarations?

17. What is type inferencing? Give an example.

18. What is the lifetime of a variable? What are the various categories of lifetimes? Give an example of how a language feature might influence the lifetime of a specific variable.

19. Given the following programming statement, describe the various bindings involved and note when they occur:
int x = 5 * y;


21. Describe how enumerated types are typically implemented. What are the limitations of such an implementation?

22. Describe at least 3 different strategies for storing strings.

23. What are the basic mechanisms that allow a programmer to create new data types and operations on that type?

24. What’s the difference between static scope and dynamic scope? Demonstrate with code how to determine each.

25. In relationship to scope, what is a hidden variable? How does it occur? Give an Ada example of creating a hidden variable, and a way of accessing it.

26. What are the relative advantages of a programming language requiring that all declarations appear at the beginning of the code, versus being able to place declarations anywhere?

27. What is the referencing environment for a given statement?

28. What are the 3 ways that the length of a string may be determined in programming languages?

29. What is a virtual origin? What purpose does it serve?

30. In general, what is the formula for accessing a typical array element, assuming row-major ordering?

31. What are the advantages of allowing programmers to define enumerated types?

32. What are the primary design issues related to defining arrays in programming languages?

33. Give an example of an associative array.

34. What are the primary design issues related to pointers?

35. What are the two biggest problems created with pointers? Give examples in code.

36. There are two ways of implementing type equivalence. What are they? Give examples. In particular, indicate how type equivalence is affected by a programming language’s choice of one type or the other, using records as a point of discussion.

37. What is the concept of “visibility”? How is visibility controlled in a typical language?