16. What is exception handling?
17. Why is readability important to writability?
18. How is the cost of compilers for a given language related to the design of that language?
19. What have been the strongest influences on programming language design over the past 50 years?
20. What is the name of the category of programming languages whose structure is dictated by the von Neumann computer architecture?
21. What two programming language deficiencies were discovered as a result of the research in software development in the 1970s?
22. What are the three fundamental features of an object-oriented programming language?
23. What language was the first to support the three fundamental features of object-oriented programming?
24. What is an example of two language design criteria that are in direct conflict with each other?
25. What are the three general methods of implementing a programming language?
26. Which produces faster program execution, a compiler or a pure interpreter?
27. What role does the symbol table play in a compiler?
28. What does a linker do?
29. Why is the von Neumann bottleneck important?
30. What are the advantages in implementing a language with a pure interpreter?

## PROBLEM SET

1. Do you believe our capacity for abstract thought is influenced by our language skills? Support your opinion.
2. What are some features of specific programming languages you know whose rationales are a mystery to you?
3. What arguments can you make for the idea of a single language for all programming domains?
4. What arguments can you make against the idea of a single language for all programming domains?
5. Name and explain another criterion by which languages can be judged (in addition to those discussed in this chapter).

6. What common programming language statement, in your opinion, is most detrimental to readability?
7. Java uses a right brace to mark the end of all compound statements. What are the arguments for and against this design?
8. Many languages distinguish between uppercase and lowercase letters in user-defined names. What are the pros and cons of this design decision?
9. Explain the different aspects of the cost of a programming language.
10. What are the arguments for writing efficient programs even though hardware is relatively inexpensive?
11. Describe some design trade-offs between efficiency and safety in some language you know.
12. In your opinion, what major features would a perfect programming language include?
13. Was the first high-level programming language you learned implemented with a pure interpreter, a hybrid implementation system, or a compiler? (You may have to research this.)
14. Describe the advantages and disadvantages of some programming environment you have used.
15. How do type declaration statements for simple variables affect the readability of a language, considering that some languages do not require them?
16. Write an evaluation of some programming language you know, using the criteria described in this chapter.
17. Some programming languages—for example, Pascal—have used the semicolon to separate statements, while Java uses it to terminate statements. Which of these, in your opinion, is most natural and least likely to result in syntax errors? Support your answer.
18. Many contemporary languages allow two kinds of comments: one in which delimiters are used on both ends (multiple-line comments), and one in which a delimiter marks only the beginning of the comment (one-line comments). Discuss the advantages and disadvantages of each of these with respect to our criteria.