1. What are the arguments for and against representing Boolean values as single bits in memory?

2. How does a decimal value waste memory space?

3. VAX minicomputers use a format for floating-point numbers that is not the same as the IEEE standard. What is this format, and why was it chosen by the designers of the VAX computers? A reference for VAX floating-point representations is Sebesta (1991).

4. Compare the tombstone and lock-and-key methods of avoiding dangling pointers, from the points of view of safety and implementation cost.

5. What disadvantages are there in implicit dereferencing of pointers, but only in certain contexts? For example, consider the implicit dereference of a pointer to a record in Ada when it is used to reference a record field.

6. Explain all of the differences between Ada's subtypes and derived types.

7. What significant justification is there for the -> operator in C and C++?

8. What are all of the differences between the enumeration types of C++ and those of Java?

9. The unions in C and C++ are separate from the records of those languages, rather than combined as they are in Ada. What are the advantages and disadvantages to these two choices?

10. Multidimensional arrays can be stored in row major order, as in C++, or in column major order, as in Fortran. Develop the access functions for both of these arrangements for three-dimensional arrays.

11. In the Burroughs Extended ALGOL language, matrices are stored as a single-dimensioned array of pointers to the rows of the matrix, which are treated as single-dimensioned arrays of values. What are the advantages and disadvantages of such a scheme?

12. Analyze and write a comparison of C's malloc and free functions with C++'s new and delete operators. Use safety as the primary consideration in the comparison.

13. Analyze and write a comparison of using C++ pointers and Java reference variables to refer to fixed heap-dynamic variables. Use safety and convenience as the primary considerations in the comparison.

14. Write a short discussion of what was lost and what was gained in Java's designers' decision to not include the pointers of C++.

15. What are the arguments for and against Java's implicit heap storage recovery, when compared with the explicit heap storage recovery required in C++? Consider real-time systems.

16. What are the arguments for the inclusion of enumeration types in C#, although they were not in the first few versions of Java?
17. What would you expect to be the level of use of pointers in C#? How often will they be used when it is not absolutely necessary?

18. Make two lists of applications of matrices, one for those that require jagged matrices and one for those that require rectangular matrices. Now, argue whether just jagged, just rectangular, or both should be included in a programming language.

19. Compare the string manipulation capabilities of the class libraries of C++, Java, and C#.

20. Look up the definition of strongly typed as given in Gehani (1983) and compare it with the definition given in this chapter. How do they differ?

21. In what way is static type checking better than dynamic type checking?

22. Explain how coercion rules can weaken the beneficial effect of strong typing?

PROGRAMMING EXERCISES

1. Design a set of simple test programs to determine the type compatibility rules of a C compiler to which you have access. Write a report of your findings.

2. Determine whether some C compiler to which you have access implements the free function.

3. Write a program that does matrix multiplication in some language that does subscript range checking and for which you can obtain an assembly language or machine language version from the compiler. Determine the number of instructions required for the subscript range checking and compare it with the total number of instructions for the matrix multiplication process.

4. If you have access to a compiler in which the user can specify whether subscript range checking is desired, write a program that does a large number of matrix accesses and time their execution. Run the program with subscript range checking and without it, and compare the times.

5. Write a simple program in C++ to investigate the safety of its enumeration types. Include at least 10 different operations on enumeration types to determine what incorrect or just silly things are legal. Now, write a C# program that does the same things and run it to determine how many of the incorrect or silly things are legal. Compare your results.

6. Write a program in C++ or C# that includes two different enumeration types and has a significant number of operations using the enumeration types. Also write the same program using only integer variables. Compare the readability and predict the reliability differences between the two programs.