depth coverage of pointers, strings, the preprocessor, structures, unions, enumerations, and low-level features of C. In addition, two chapters (15 and 19) offer guidance on program design.

- ***The Standard C Library.*** Chapters 21–27 focus on the C library, a large collection of functions that come with every compiler. These chapters are most likely to be used as reference material, although portions are suitable for lectures.
- ***Reference.*** Appendix A gives a complete list of C operators. Appendix B describes the major differences between C99 and C89, and Appendix C covers the differences between C89 and K&R C. Appendix D is an alphabetical listing of all functions in the C89 and C99 standard libraries, with a thorough description of each. Appendix E lists the ASCII character set. An annotated bibliography points the reader toward other sources of information.

A full-blown course on C should cover Chapters 1–20 in sequence, with topics from Chapters 21–27 added as needed. (Chapter 22, which includes coverage of file input/output, is the most important chapter of this group.) A shorter course can omit the following topics without losing continuity: Section 8.3 (variable-length arrays), Section 9.6 (recursion), Section 12.4 (pointers and multidimensional arrays), Section 12.5 (pointers and variable-length arrays), Section 14.5 (miscellaneous directives), Section 17.7 (pointers to functions), Section 17.8 (restricted pointers), Section 17.9 (flexible array members), Section 18.6 (inline functions), Chapter 19 (program design), Section 20.2 (bit-fields in structures), and Section 20.3 (other low-level techniques).

Exercises and Programming Projects

Having a variety of good problems is obviously essential for a textbook. This edition of the book contains both exercises (shorter problems that don't require writing a full program) and programming projects (problems that require writing or modifying an entire program).

A few exercises have nonobvious answers (some individuals uncharitably call these "trick questions"—the nerve!). Since C programs often contain abundant examples of such code, I feel it's necessary to provide some practice. However, I'll play fair by marking these exercises with an asterisk (*). Be careful with a starred exercise: either pay close attention and think hard or skip it entirely.

Errors, Lack of (?)

I've taken great pains to ensure the accuracy of this book. Inevitably, however, any book of this size contains a few errors. If you spot one, please contact me at *cbook@knking.com*. I'd also appreciate hearing about which features you found especially helpful, which ones you could do without, and what you'd like to see added.