- *Frequently asked questions.* I've tried to answer questions that come up frequently in my own courses, in other books, and on newsgroups related to C.
- *Additional discussion and clarification of tricky issues.* Although readers with experience in a variety of languages may be satisfied with a brief explanation and a couple of examples, readers with less experience need more.
- *Side issues that don't belong in the main flow.* Some questions raise technical issues that won't be of interest to all readers.
- *Material too advanced or too esoteric to interest the average reader.* Questions of this nature are marked with an asterisk (*). Curious readers with a fair bit of programming experience may wish to delve into these questions immediately; others should definitely skip them on a first reading. *Warning:* These questions often refer to topics covered in later chapters.
- *Common differences among C compilers.* I discuss some frequently used (but nonstandard) features provided by particular compilers.

Q&A

Some questions in Q&A sections relate directly to specific places in the chapter; these places are marked by a special icon to signal the reader that additional information is available.

Other Features

In addition to Q&A sections, I've included a number of useful features, many of which are marked with simple but distinctive icons (shown at left).

⚠

- **Warnings** alert readers to common pitfalls. C is famous for its traps; documenting them all is a hopeless—if not impossible—task. I've tried to pick out the pitfalls that are most common and/or most important.

cross-references ➤*Preface*

- **Cross-references** provide a hypertext-like ability to locate information. Although many of these are pointers to topics covered later in the book, some point to previous topics that the reader may wish to review.

idiom

- **Idioms**—code patterns frequently seen in C programs—are marked for quick reference.

portability tip

- **Portability tips** give hints for writing programs that are independent of a particular machine, compiler, or operating system.
- **Sidebars** cover topics that aren't strictly part of C but that every knowledgeable C programmer should be aware of. (See "Source Code" on the next page for an example of a sidebar.)
- **Appendices** provide valuable reference information.

Programs

Choosing illustrative programs isn't an easy job. If programs are too brief and artificial, readers won't get any sense of how the features are used in the real world. On the other hand, if a program is *too* realistic, its point can easily be lost in a forest of