

details. I've chosen a middle course, using small, simple examples to make concepts clear when they're first introduced, then gradually building up to complete programs. I haven't included programs of great length; it's been my experience that instructors don't have the time to cover them and students don't have the patience to read them. I don't ignore the issues that arise in the creation of large programs, though—Chapter 15 (Writing Large Programs) and Chapter 19 (Program Design) cover them in detail.

I've resisted the urge to rewrite programs to take advantage of the features of C99, since not every reader may have access to a C99 compiler or wish to use C99. I have, however, used C99's `<stdbool.h>` header in a few programs, because it conveniently defines macros named `bool`, `true`, and `false`. If your compiler doesn't support the `<stdbool.h>` header, you'll need to provide your own definitions for these names.

The programs in this edition have undergone one very minor change. The `main` function now has the form `int main(void) { ... }` in most cases. This change reflects recommended practice and is compatible with C99, which requires an explicit return type for each function.

---

---

### *Source Code*

Source code for all programs is available at [knking.com/books/c2](http://knking.com/books/c2). Updates, corrections, and news about the book can also be found at this site.

---

---

### **Audience**

This book is designed as a primary text for a C course at the undergraduate level. Previous programming experience in a high-level language or assembler is helpful but not necessary for a computer-literate reader (an “adept beginner,” as one of my former editors put it).

Since the book is self-contained and usable for reference as well as learning, it makes an excellent companion text for a course in data structures, compiler design, operating systems, computer graphics, embedded systems, or other courses that use C for project work. Thanks to its Q&A sections and emphasis on practical problems, the book will also appeal to readers who are enrolled in a training class or who are learning C by self-study.

### **Organization**

The book is divided into four parts:

- *Basic Features of C.* Chapters 1–10 cover enough of C to allow the reader to write single-file programs using arrays and functions.
- *Advanced Features of C.* Chapters 11–20 build on the material in the earlier chapters. The topics become a little harder in these chapters, which provide in-