

Preface

C is one of the most popular programming languages used in engineering and science. It is the language of choice for many engineers and scientists. This book is designed to teach students how to solve engineering and science problems using the C programming language. The book teaches beginners with no previous programming experience the underlying working principles of scientific computing and a disciplined approach for software development. All the major features of C89 and C99 are presented with numerous engineering application examples derived from production code. The book reveals the coding techniques used by the best C programmers and shows how experts solve problems in C. It can also serve as a reference book for seasoned programmers.

Prerequisites

The mathematical prerequisite for the book is trigonometry, except for Chapters 22 and 23 (section 23.11), which require a calculus background. No prior computer or programming experience is required. However, if students have experience in other programming languages, the material in the first few chapters can be covered quickly. For upper-division students, discussions on number systems and the introduction on vectors and matrices can be skipped.

Special Features

This book grew out of my sixteen years of teaching introductory computer programming in C for engineering applications and engineering software design at the University of California, Davis since 1992. It also reflects my unique experience in the design and implementation of a cross-platform C/C++ interpreter Ch as well as my extensive industrial and research work on solving engineering problems in C and C++. The book is written with the following distinct features.

- *Complete C coverage.* This book completely covers the C language in Part I, which is organized cumulatively so that each chapter builds on information presented in the earlier chapters. A single topic is presented in a section first and then reinforced and cross-referenced in other sections and chapters. The text is logically organized and heavily cross-referenced and indexed to serve as a reference for serious programmers.
- *Extensive coverage of major C99 features.* C99 features four variable length arrays (VLA), type generic functions, IEEE 754 floating-point arithmetic, complex numbers, and new data types `bool` and `long long`, which are used for numerical computing. They overcome the shortcomings of C89 for solving problems in engineering and science.
- *Plotting for visualization.* A picture is worth a thousand words. A plotting C++ library is used for visualization for solving many problems in engineering and science in both the text and exercises. All plots including those in exercises in this book were generated using simple functions or member functions in this plotting library.

- *Cross-platform software development.* All code examples in Parts I and II have been tested in Windows, Linux, Mac OS X, Solaris, HP-UX, FreeBSD, and QNX in C/C++ compilers and cross-platform C/C++ interpreter Ch. The implementation and platform-specific issues are presented with examples throughout the book in sections titled “**Making It Work**” at the end of each chapter in Parts I and II. Building and distributing static and dynamic libraries using make files on these platforms is also demonstrated.
- *64-bit programming.* Whenever the code produces different results for 32-bit and 64-bit machines, it is pointed out in the text. All results have been tested in multiple platforms using C/C++ interpreter and compilers.
- *Extensive coverage and unique presentation of difficult concepts.* Difficult topics such as arrays, pointers and many related subjects, binary and text mode for accessing files, byte order (endianness), structure alignment, dynamic data structures, static and dynamic libraries, internal representation of floating-point numbers, and hardware interfaces are thoroughly covered using multiple forms of presentation with many sample application programs.
- *LAPACK (Linear Algebra PACKage) written in Fortran and its C version, CLAPACK, are introduced.* Mixed-language programming with both row-wise and column-wise arrays is addressed.
- *Commonly used Unix commands are used for cross-platform software development.* In addition, the input/output redirection is introduced at the beginning for using data files in programs.

In addition, the companion CD contains Ch, a cross-platform C/C++ interpreter, with a complete reference manual in a PDF file for all functions in the C standard libraries. I originally designed and implemented Ch to make my job of teaching and research easier. As the user base increased, Ch evolved from a special-application program to a general-purpose computing environment with wide applicability. Ch was never meant to be a new language. Conforming to the C standard is its prevailing design goal. As a complete C interpreter, Ch conforms to the C89 standard and supports major features in C99. C expressions, statements, functions, and programs can be executed in Ch interactively without compiling and linking the code. It can be used for quick testing and trying difficult C features such as pointers and arrays, especially for classroom presentation and discussion sessions. ChIDE, a user-friendly Integrated Development Environment (IDE), can be used to execute functions and programs with detailed line-by-line traces. The Ch environment for Windows contains nearly 250 commonly used Unix commands, such as **vi**, **ls**, **rm**, **awk**, **sed**, **gzip**, and **tar**. It can also be used by students to learn Unix in a familiar Windows environment for a smooth transition from Windows to Unix. Ch has been well received by both instructors and students. In this book, many succinct C code and statements are executed in Ch interactively with output to illustrate numerous programming features and concepts.

Furthermore, many illustrations are used to clarify difficult concepts for beginners. Flowcharts, pseudocode, and procedures are used to describe complicated algorithms. The disciplined approach, various features of C, programming style, modular programming, code reuse, and algorithm development for solving problems in engineering and science are illustrated by over 350 well-documented complete sample programs consisting of more than 14,000 lines of code. Many of these programs are derived from the real-world production code. Earlier example programs are gradually rewritten to make them more concise, efficient, powerful, useful, or user-friendly. This way, students do not struggle with trying to understand new features and new programs at the same time. The top-down and bottom-up software design and refinement are presented with large-scale application examples.

Finally, over 500 carefully designed exercises for solving many problems in engineering and science reinforce concepts presented in the text. Abundant exercises are designed for teaching multiple sessions without repeating problems.

Organization of the Book

The text is divided into four parts. Part I is about structured programming in C, which is the major focus of the book.

Part II moves from structured programming to object-based programming in C++. It presents a C++ class for graphical plotting.

Part III introduces computational arrays as first-class objects and advanced numerical functions in Ch. They can be used conveniently to solve linear systems, nonlinear equations, ordinary differential equations, etc., for problems in engineering and science. The concepts presented in Part III are applicable to other numerically oriented programming languages such as Fortran and MATLAB.

Part IV introduces array-based numerical computing in MATLAB as a second programming language compared to C. It may serve as a quick reference for programming in MATLAB for C programmers.

The chapter on an introduction to Fortran as a second programming language in comparison with C can be downloaded in a PDF file from the website for the book. The comparison study presented in this chapter will also be very useful for those who have the prior programming experience in Fortran to learn C.

Appendix A lists keywords in C and Ch. Appendix B lists C99 features supported in Ch. Appendix C lists C++ features supported in Ch. Appendix D lists ASCII codes.

Sections and chapters marked with the double dagger symbol ‘‡’ in the title can be skipped without hindering learning about the later chapters. The materials marked with the symbol ‘‡’, however, are very useful for those who plan on serious software development in C, need to read existing C code written by experienced programmers, or intend to take other advanced courses.

Part I is organized cumulatively except for Chapter 16, “Scientific Computing in the Entire Real Domain in C99,” and Chapter 17, “Programming with Complex Numbers in C99 and C++.” These two chapters can be read after Chapter 11, “Pointers,” is finished. Other parts are self-contained. After covering sections not marked with the symbol ‘‡’ in Part I, one can move to Part II. Likewise, after finishing Chapter 10, “Arrays,” in Part I, one can continue to Part III or Part IV.

Using this Book as a Textbook

This is a comprehensive book on software development for solving problems in engineering and science with complete coverage of the C language and other programming languages. The materials are more than enough for a one-semester course. Early drafts of this book have been used as a textbook for a number of courses. Following are some possible ways to use the book as a textbook.

- **Computer Programming for Engineering Applications or Engineering Problem Solving.** In this freshman introductory course, it is assumed that students have no prior computer and programming experience. All sections and chapters not marked with the symbol ‘‡’ are covered in one quarter. It takes eight weeks to cover Chapters 1 to 14 in Part I, one week on computational arrays in Chapter 21 (in Part III) and plotting in Chapter 20 (in Part II), and one week on Chapter 23 (Part IV) for comparison study with MATLAB. If it is a semester course, additional topics marked with the symbol ‘‡’ may be covered. Experience indicates that students who have C can pick up MATLAB quickly.
- **Introduction to Programming in C.** The book can be used as a gentle, slow-paced introduction to computer programming using the C language. It is assumed that students do not have any prior computer or programming experience. Example programs can be presented using ChIDE with detailed line-by-line traces. In this case, only sections in Chapters 1 to 10 not marked with the symbol ‘‡’ and the first two sections in Chapter 14 will be covered.

- **Introduction to C Programming.** For this upper-division undergraduate course, it is assumed that students have prior programming experience in other programming languages or software packages. The materials in early chapters are presented much faster. The most time is spent on Chapters 1 to 14 in Part I. It takes one-and-a-half weeks to cover numerical computing in Ch in Part III and plotting in Chapter 20 (in Part II).
- **Using this book as a supplementary textbook.** The text can also be used as a supplement for some upper-division or graduate courses in engineering and science, such as numerical analysis, control, mechatronics, embedded systems, mechanism design, and engineering software design, in which C programming is needed.

C89 and C99 Standards

There are two versions of the international standards for C. The first version was ratified in 1989 by the American National Standards Institute (ANSI) C standard committee. It is often referred as ANSI C or C89. The second C standard was completed in 1999. This standard is commonly referred to as C99. C99 is a milestone in C's evolution into a viable programming language for numerical and scientific computing. It can significantly simplify programming tasks for solving problems in engineering and science. An increasing number of C compilers support C99 features. Major numerical features in C99 are presented in this book for applications in engineering and science. However, Microsoft Visual C++ still does not support major features in C99 for numerical computing. All programs using C99 features have been tested with both the GNU gcc compiler in Linux and the cross-platform C/C++ interpreter Ch. If a C99 feature is used, it will be pointed out explicitly in both the text and title of the section or chapter. Programs using C99 features have a comment with the compilation option `-std=c99` at the beginning to distinguish from those using C89 features only. For example, the program `funcname.c` in Program 6.16 in Chapter 6 contains the comment

```
Compile in gcc, use command 'gcc -std=c99 funcname.c'
```

Websites for the Book and Teaching Resources

The McGraw-Hill website for the book is

<http://www.mhhe.com/cheng>

Many teaching resources are available at no cost from this website to instructors who adopt this book as a textbook for their classes. They include 1,400 PowerPoint slides complementary to this book with many different example programs ready for classroom presentations, over 200 companion PowerPoint slides for discussion sessions, *Instructor's Guide* (containing sample general information, syllabus, homework assignments, quizzes, and midterm and final examinations), and *Solutions Manual* and the source code for solutions for exercises in each chapter. The site also contains useful resources for students, including a chapter on Fortran as a second language in comparison study with C in a PDF file.

The author's website for the book is

<http://iel.ucdavis.edu/cfores>

This website contains the source code for all example programs presented in the book, frequently asked questions, supplementary materials, errata, and numerous other resources. The first line of each example program contains its file name so that it can be found easily in the downloaded source code. The website also has a link to a discussion forum for students to exchange ideas and share learning experiences.

xvi Preface

Contacting the Author

I sincerely appreciate any criticisms, comments, identification of errors in the text or programs, and suggestions for improvement in future editions from both instructors and students. I can be reached over the Internet at

cfores@ucdavis.edu

I will post corrections and clarifications on the author's website for the book.

Electronic Textbook Options

This text is offered through CourseSmart to both instructors and students. CourseSmart is an online resource where students can purchase access to this and other McGraw-Hill textbooks in a digital format. Through their browsers, students can access the complete text online for one year at almost half the cost of a traditional text. Purchasing the eTextbook also allows students to take advantage of CourseSmart's Web tools for learning, which include full text search, notes and highlighting, and e-mail tools for sharing notes between classmates. To learn more about CourseSmart options, contact your sales representative or visit *www.CourseSmart.com*.

Acknowledgments

First and foremost, I would like to thank the numerous undergraduate and graduate students who used drafts of this book to learn computer programming for solving engineering problems and software design. Their feedback and suggestions greatly improved the text. I also would like to acknowledge my former and current graduate students, especially Bo Chen and Stephen Nestinger, for their numerous thoughtful comments and suggestions.

The insightful criticisms and suggestions of many reviewers have greatly improved the accuracy and presentation of this book. I would like to acknowledge their efforts. They include David Auslander (University of California-Berkeley), Rebecca Bates (Minnesota State University-Mankato), Thomas Daniels (Iowa State University), Paul Furth (New Mexico State University), Tom MacDonald (Eivia, Inc.), Gary Nutt (University of Colorado-Boulder). I am especially grateful to Tom MacDonald, David Auslander, and Paul Furth for scrutinizing the text and programs with countless suggestions. However, I take the full responsibility for any errors. I also would like to thank Paul Furth for providing several exercise problems in Chapter 10.

The outstanding team at McGraw-Hill has been delightful to work with on this project. It includes Raghu Srinivasan, Lorraine Buczek, and Melissa Leick, Curt Reynolds, Laura Lawrie, and Susan McClung. I appreciate their extraordinary efforts in publishing this book.

Finally, I would like to thank my wife, Dawn Cheng, and daughter, Deanna Cheng, for their support and understanding over the many years during which this book has been developed and written.

Harry H. Cheng
November 20, 2008