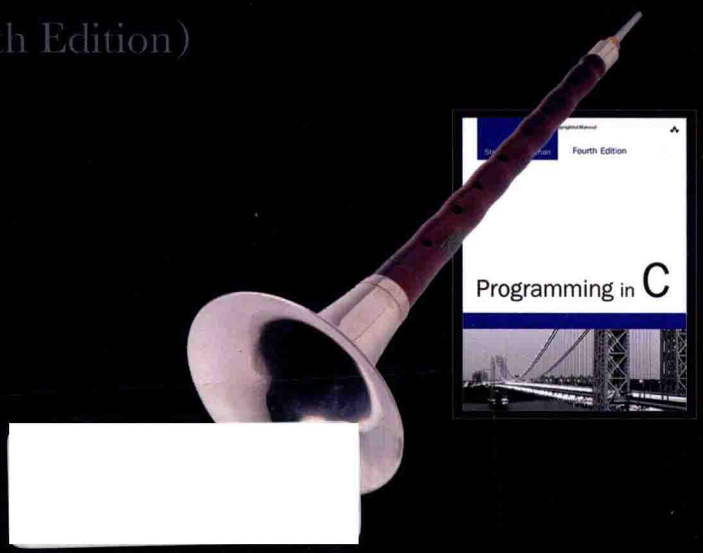
Broadview
www.broadview.com.cn
PEARSON
· 原味精品书系

C语言程序设计（英文版）
（第4版）

Programming in C（4th Edition）

[美] Stephen G. Kochan 著



中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

· 原味精品书系 ·

C语言程序设计（英文版）（第4版）

Programming in C（4th Edition）

[美] Stephen G. Kochan 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内容简介

本书全面介绍了C语言的各种特性，包括C11中增加的内容。书中包含大量完整的示例及详细的讲解。附录中详尽总结了C语言和C语言库，两者的组织形式都便于快速参考。

本书通过示例来教授C语言，每个新概念都有完整的C程序做演示。你不仅可以学习这门语言的基础知识，还能养成良好的程序设计习惯。另外，每章最后附有习题，便于课堂学习或自学。

无论是否拥有编程经验，你都可以通过本书透彻地理解C语言。

版权贸易合同登记号　图字：01-2015-5599

图书在版编目（CIP）数据

C语言程序设计：第4版 =Programming in C，4E：英文 /（美）寇肯（Kochan,S.G.）著. — 北京：电子工业出版社，2016.4
（原味精品书系）
ISBN 978-7-121-27319-3
I. ① C…II. ① 寇…III. ① C语言－程序设计－英文 IV. ① TP312
中国版本图书馆CIP数据核字（2015）第234566号

策划编辑：张春雨　刘　芸
责任编辑：徐津平
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×980 1/16　　印张：33.75　　字数：650千字
版　　次：2016年4月第1版
印　　次：2016年4月第1次印刷
定　　价：108.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
服务热线：（010）88258888。

献给我的父亲和母亲

关于作者

Stephen G. Kochan 使用C语言设计软件已经有超过30年的时间。他是几本最畅销的C语言书籍的作者，包括 *Programming in C*、*Programming in Objective-C* 和 *Topics in C Programming*。他在UNIX方面也有大量著述，是 *Exploring the UNIX System* 和 *UNIX Shell Programming* 等书的著者或合著者。

对本书有贡献的作者

Dean Miller 是一位作家和编辑，在出版和特许消费者产品领域都拥有超过20年的经验。他是 *Sams Teach Yourself C in One Hour a Day* 和 *Sams Teach Yourself Beginning Programming in 24 Hours* 最近几个版本的合著者。

致谢

衷心地感谢下列各位在我准备本书各个版本时提供的帮助：Douglas McCormick、Jim Scharf、Henry Tabickman、Dick Fritz、Steve Levy、Tony Ianinno 和 Ken Brown。还要感谢纽约大学的 Henry Mullish 在写作方面给我的诸多教益，感谢他带我走进出版领域。

在 Pearson 的工作人员中，我要特别感谢 Mark Taber 和责任编辑 Mandie Frank，还要感谢文字编辑 Charlotte Kughen 和技术编辑 Siddhartha Singh。最后，在 Pearson 还有许多参与本项目但没有与我直接接触的工作人员，在此向他们所有人表示感谢。

目录

Introduction

The C programming language was pioneered by Dennis Ritchie at AT&T Bell Laboratories in the early 1970s. It was not until the late 1970s, however, that this programming language began to gain widespread popularity and support. This was because until that time C compilers were not readily available for commercial use outside of Bell Laboratories. Initially, C's growth in popularity was also spurred on in part by the equal, if not faster, growth in popularity of the Unix operating system. This operating system, which was also developed at Bell Laboratories, had C as its "standard" programming language. In fact, well over 90% of the operating system itself was written in the C language!

The enormous success of the IBM PC and its look-alikes soon made MS-DOS the most popular environment for the C language. As C grew in popularity across different operating systems, more and more vendors hopped on the bandwagon and started marketing their own C compilers. For the most part, their version of the C language was based on an appendix found in the first C programming text—*The C Programming Language*—by Brian Kernighan and Dennis Ritchie. Unfortunately, this appendix did not provide a complete and unambiguous definition of C, meaning that vendors were left to interpret some aspects of the language on their own.

In the early 1980s, a need was seen to standardize the definition of the C language. The American National Standards Institute (ANSI) is the organization that handles such things, so in 1983 an ANSI C committee (called X3J11) was formed to standardize C. In 1989, the committee's work was ratified, and in 1990, the first official ANSI standard definition of C was published.

Because C is used around the world, the International Standard Organization (ISO) soon got involved. They adopted the standard, where it was called ISO/IEC 9899:1990. Since that time, additional changes have been made to the C language. The most recent standard was adopted in 2011. It is known as ANSI C11, or ISO/IEC 9899:2011. It is this version of the language upon which this book is based.

C is a "higher-level language," yet it provides capabilities that enable the user to "get in close" with the hardware and deal with the computer on a much lower level. This is because, although C is a general-purpose structured programming language, it was originally designed with systems programming applications in mind and, as such, provides the user with an enormous amount of power and flexibility.

This book proposes to teach you how to program in C. It assumes no previous exposure to the language and was designed to appeal to novice and experienced programmers alike. If you have previous programming experience, you will find that C has a unique way of doing things that probably differs from other languages you have used.

Every feature of the C language is treated in this text. As each new feature is presented, a small *complete* program example is usually provided to illustrate the feature. This reflects the overriding philosophy that has been used in writing this book: to teach by example. Just as a picture is worth a thousand words, so is a properly chosen program example. If you have access to a computer that supports the C programming language, you are strongly encouraged to download and run each program presented in this book and to compare the results obtained on your system to those shown in the text. By doing so, not only will you learn the language and its syntax, but you will also become familiar with the process of typing in, compiling, and running C programs.

You will find that program readability has been stressed throughout the book. This is because I strongly believe that programs should be written so that they can be easily read—either by the author or by somebody else. Through experience and common sense, you will find that such programs are almost always easier to write, debug, and modify. Furthermore, developing programs that are readable is a natural result of a true adherence to a structured programming discipline.

Because this book was written as a tutorial, the material covered in each chapter is based on previously presented material. Therefore, maximum benefit will be derived from this book by reading each chapter in succession, and you are highly discouraged from "skipping around." You should also work through the exercises that are presented at the end of each chapter before proceeding on to the next chapter.

Chapter 1, "Some Fundamentals," which covers some fundamental terminology about higher-level programming languages and the process of compiling programs, has been included to ensure that you understand the language used throughout the remainder of the text. From Chapter 2, "Compiling and Running Your First Program," on, you will be slowly introduced to the C language. By the time Chapter 15, "Input and Output Operations in C," rolls around, all the essential features of the language will have been covered. Chapter 15 goes into more depth about I/O operations in C. Chapter 16, "Miscellaneous and Advanced Features," includes those features of the language that are of a more advanced or esoteric nature.

Chapter 17, "Debugging Programs," shows how you can use the C preprocessor to help debug your programs. It also introduces you to interactive debugging. The popular debugger `gdb` was chosen to illustrate this debugging technique.

Over the last decade, the programming world has been abuzz with the notion of object-oriented programming, or OOP for short. C is not an OOP language; however, several other programming languages that are based on C are OOP languages. Chapter 18, "Object-oriented Programming," gives a brief introduction to OOP and some of its terminology. It also gives a brief overview of three OOP languages that are based on C, namely C++, C#, and Objective-C.

Appendix A, "C Language Summary," provides a complete summary of the language and is provided for reference purposes.

Appendix B, "The Standard C Library," provides a summary of many of the standard library routines that you will find on all systems that support C.

Appendix C, "Compiling Programs with `gcc`," summarizes many of the commonly used options when compiling programs with GNU's C compiler `gcc`.

In Appendix D, "Common Programming Mistakes," you'll find a list of common programming mistakes.

Finally, Appendix E, "Resources," provides a list of resources you can turn to for more information about the C language and to further your studies.

This book makes no assumptions about a particular computer system or operating system on which the C language is implemented. The text makes brief mention of how to compile and execute programs using the popular GNU C compiler `gcc`.