

C11编程导论

(英文版)

[美] Paul Deitel 著
Harvey Deitel

Deitel (TM)

C for Programmers
with an
Introduction
to C11



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

· 原味精品书系 ·

C11 编程导论 (英文版)

C for Programmers with an Introduction to C11

[美] Paul Deitel 著
Harvey Deitel



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书沿用了 Deitel 特色的“程序实况解说”，深入探讨了 C 语言和 C 标准库。通过完整的语法着色、代码高亮、代码演练和程序输出，全面地介绍了测试程序的概念。本书汇集约 5000 行 C 代码和数百个开发技巧，将帮助你构建强大的应用程序。本书的内容包括构建自定义数据结构、标准库等；并选择了 C11 标准的一些新特性，如多线程来帮助你为目前的多核系统编写高性能的应用程序；C 语言安全编程部分展示了如何让你编写的程序具有更好的鲁棒性，不易受到攻击。

本书适合具有一定高级语言编程背景的程序员阅读。

Original edition, entitled C for Programmers with an Introduction to C11, 1E, 0133462064 by Paul Deitel and Harvey Deitel, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright©2013 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by Pearson Education Asia Ltd. and Publishing House of Electronics Industry Copyright © 2016. The edition is manufactured in the People's Republic of China, and is authorized for sale and distribution only in the mainland of China exclusively (except Hong Kong SAR, Macau SAR, and Taiwan).

本书英文影印版专有出版权由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书仅限中国大陆境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

本书英文影印版贴有 Pearson Education 培生教育出版集团激光防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2015-6097

图书在版编目 (CIP) 数据

C11 编程导论 = C for Programmers with an Introduction to C11, 1E: 英文 / (美) 戴特尔 (Deitel, P.), (美) 戴特尔 (Deitel, H.) 著. — 北京: 电子工业出版社, 2016.4
(原味精品书系)

ISBN 978-7-121-27316-2

I. ① C… II. ① 戴… ② 戴… 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 233678 号

责任编辑: 张 玲

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

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: 29.75 字数: 571 千字

版 次: 2016 年 4 月第 1 版

印 次: 2016 年 4 月第 1 次印刷

定 价: 89.00 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。

前言

欢迎大家和我一起学 C 语言。本书会向软件开发专业人员介绍最先进的计算技术。

本书沿用了 Deitel 特色的“程序实况解说”风格，我们会用完整的可用性程序（而不是代码片段）来解释相关概念。每个代码示例后都跟着一个或者多个样例。请在线阅读“Before You Begin”的内容（http://www.deitel.com/bookresources/cfp/cfp_byb.pdf），并据此设置自己的电脑，以便顺畅地运行这 130 个代码示例和你自己的 C 程序。所有的源代码都可以从 www.deitel.com/books/cfp 和 www.pearsonhighered.com/deitel 下载。当学习这些程序时，请使用提供的源代码来运行它们。

本书在详细介绍 C 语言的同时，会兼顾挑战性和趣味性。如果有任何问题，都可以发送邮件到 deitel@deitel.com，我们会及时回复。要了解本书的最新信息，可以访问 www.deitel.com/books/cfp，或加入 Facebook 社区（www.deitel.com/deitelfan）、Twitter 社区（@deitel）、Google+ 社区（[gplus.to/deitel](https://plus.google.com/deitel)）和 LinkedIn 社区（bit.ly/deitelLinkedIn）。此外，你还可以订阅 *Deitel Buzz Online* 电子邮件新闻简报（www.deitel.com/newsletter/subscribe.html）来了解更多相关内容。

本书特色

本书的主要特色如下：

- **覆盖 C 语言新标准。**本书是针对 2011 年批准的 C 语言新标准编写的，该标准通常被称为 C11 或者被简单地称为“C 标准”。不同编译器对新标准的支持有所不同。我们的大多数读者会选择使用 GNU gcc 编译器（支持新标准中的很多关键特性）或者微软的 Visual C++ 编译器。微软只支持被添加到 C99 和 C11 中的一小部分 C 特性——主要是 C++ 标准中也需要的那些特性。为了适应所有的读者，我们将新标准特性的讨论放在易于使用的相关章节，以及附录 E “C 语言新标准的附加特性”中。我们还用由 C 语言新标准引起的首选版本取代了过时的功能。
- **C 语言安全编程。**我们在很多 C 语言编程章节中都添加了关于 C 语言安全编程的注意事项。我们还在 www.deitel.com/SecureC/ 上发布了 C 语言安全编程资源中心。要了解更多细节，请参见下面的“C 语言安全编程注意事项”的相关内容。
- **专注于性能问题。**C 语言经常受到性能密集型应用程序（例如，操作系统、实时系统、嵌入式系统和通信系统）设计师的青睐，因此，我们会密切关注性能问题。

- 所有的代码都在 Windows 和 Linux 下进行了测试。我们分别使用 Visual C++ (Windows) 和 GNU gcc (Linux) 对每个示例代码进行了测试。
- 更深入地研究了排序。排序是一个很有趣的问题,因为虽然不同的排序算法最终会得到相同的结果,但它们的内存消耗、占用CPU的时间和和其他系统资源完全不同——算法性能至关重要。我们在第6章开始介绍排序,并且在附录D中对其进行了更深入的探讨。我们考虑了几种算法,并从它们的内存消耗和对处理器的要求方面进行了比较。为此,我们引入了大O表示法,它用于表示使用算法解决问题的困难程度。我们在附录D讨论了选择排序、插入排序和递归归并排序。
- 调试器附录。本书中包含了 Visual Studio 和 GNU gdb 调试附录。
- 运算顺序。本书讨论了运算顺序问题,来帮助你避免错误。
- C++ 风格的 // 注释。我们使用的是较新的、更简洁的 C++ 风格 // 注释,而不是老式的 C 风格 /*...*/ 注释。
- C 标准库。我们在 1.3 节引用了 en.cppreference.com/w/c, 在这里你可以找到 C 标准库函数的详细可搜索文件。

C 语言安全编程注意事项

经验表明,人们很难创建出可以抵抗病毒、蠕虫等攻击的具有工业强度的系统。如今,通过互联网,这类攻击瞬间就可以对全球范围造成影响,而这些软件漏洞往往来自很容易避免的编程问题。从开发周期的开始就将安全性考虑进来,可以大大减少开发的成本和漏洞。

人们创建了 CERT Coordination Center (www.cert.org) 来分析和及时应对攻击。CERT (计算机安全应急响应组, Computer Emergency Response Team) 标准会被发布和推广,以帮助 C 语言程序员和其他人实现具有工业强度系统的安全编码标准,这样就可以避免造成开放式系统受到攻击的编程实践。CERT 标准将随着新出现的安全问题而不断演变。

我们的代码符合书籍水平的 CERT 建议。如果你要创建在工业上应用的 C 语言系统,可以考虑阅读 Robert Seacord 的《C 安全编码标准》(Addison-Wesley Professional, 2009)、《C 和 C++ 安全编码》(Addison-Wesley Professional, 2013), 读者可以在 www.securecoding.cert.org 上免费在线阅读 CERT 指南。本书的技术评审 Seacord 也为新的 C 语言安全编程部分提供了具体的建议。Seacord 先生是卡内基梅隆大学软件工程研究所 (SEI) CERT 的安全编程管理员,他还是卡内基梅隆大学计算机科学系的副教授。

我们在第2章~第13章中各章的最后一节 Secure C Programming (C 语言安全编程) 部分讨论了很多重要的主题,包括运算溢出的测试、无符号整数类型的使用、C 标准附录 K 中的新安全函数、检查标准库函数返回的状态信息的重要性、范围检查、安全随机数的生成、数组边界检查、避免缓冲区溢出的技术、输入验证、避免未定义的行为、返回状态信息的函数和未返回状态信息的类似函数的选择、确保指针总是 NULL 或者包含有效地址、预处理宏的首选 C 函数等。

教学方法

本书包含丰富的示例。我们将专注于良好的软件工程并强调程序的清晰性。

语法着色。为了增加可读性，我们为代码添加了语法阴影，这与大多数 IDE 和代码编辑器给代码的语法添加颜色是类似的。我们采用的语法着色约定是：

注释用常规字体表示。

关键字用粗字体表示。

常量和字面值用比关键字稍浅一点的黑体字来表示。

所有的其他代码用黑体表示。

代码高亮显示。我们在每个源代码程序的关键代码段都放置了一个灰色的矩形框。

使用字体突出显示。为了便于参考，我们用粗体表示每个定义性出现的关键概念和索引的页面引用。我们用加粗的 Helvetica 字体来强调屏幕组件（例如，**File** 菜单）、用 Lucida 字体来强调 C 语言程序文本（例如，`int x = 5;`）。

目标。每章都包含一个 Objectives（本章目标）列表。

插图 / 图。我们在本书中包含了大量的图表、表格、线条图、流程图、程序和程序输出。

编程技巧。我们提供了编程技巧来帮助你关注程序开发的重要方面。这些技巧和实践是我们在 8 年编程和企业培训经验中积累的精华。



最佳编程实践

最佳编程实践让读者关注那些有助于编写更清晰、更易理解和维护的程序的技术。



常见编程错误

指出这些常见的编程错误可以减少读者犯类似错误的可能性。



错误预防技巧

这些技巧包括揭露和删除程序中 bug 的建议；大多数技巧描述的是如何预防将 bug 带入 C 程序中。



性能技巧

这些技巧强调的是如何让程序运行得更快或者尽量减少它们占用的内存。



可移植性技巧

可移植性技巧可以帮助你编写能够在各种平台上运行的代码。



软件工程意见

软件工程意见强调的是影响软件系统（尤其是大型系统）构建的架构和设计问题。

索引。我们在书中包含了大量的索引，它在你将本书当作参考书使用时尤为有用。我们使用加粗的页号来强调关键术语的定义出现的页码。

本书使用的软件

我们在本书中使用的是免费的 GNU C 编译器 (gcc.gnu.org/install/binaries.html)，其中大多数 Linux 系统都已经预先安装了该编译器，它还可以被安装到 Mac OS X、Windows 系统和用于 Windows 桌面的免费 Visual Studio Express 2012 (www.microsoft.com/express) 中。Apple 的 Xcode 开发工具中包含 LLVM 编译器，Mac OS X 用户可以从 Mac App Store 中免费下载。很多其他免费的 C 编译器都可以在线获取。

C 基础：LiveLessons 视频培训产品的第 I 部分和第 II 部分

“C 基础：LiveLessons 视频培训产品的第 I 部分和第 II 部分”（2013 年秋季上线）介绍了开始使用 C 语言构建健壮的、强大的软件时需要了解的内容。它包含了十多个小时的与本书同步的专家培训视频。要了解关于 Deitel LiveLessons 视频产品的更多信息，请访问

www.deitel.com/livelessons

或者通过 deitel@deitel.com 联系我们。如果你已经订阅了 Safari Books Online (www.safaribooksonline.com)，你还可以获取我们的 LiveLessons 视频。

致谢

我们要感谢 Abbey Deitel 和 Barbara Deitel，他们为这个项目投入了大量时间。我们有幸与 Prentice Hall/Pearson 的出版专家团队共同完成这个项目。我们要感谢培生科技出版集团的主编 Mark L. Taub 17 年来给我们的指导和为此付出的精力。我们要感谢 Carole Snyder 出色地完成了管理审稿过程的工作，感谢 Chuti Prasertsith 为本书设计了具有创造力和精确度的封面，感谢 John Fuller 出色地完成了我们的 Deitel 开发人员系列图书的出版工作。

审稿人员

我们要感谢本书审稿人员的付出，他们在严格的最终期限的压力下完成了对本书文本和程序的审查，并为提高本书的表述方式提供了无数建议，他们是：John F. Doyle 博士（印第安纳大学东南分校）、Hemanth H.M.（SonicWALL 的软件工程师）、Vytautas Leonavivius（微软）、Robert Seacord（SEI/CERT 的安全编码管理员、《C 安全编码标准》一书的作者、C 编程语言国际标准工作组的技术专家）和 José Antonio González Seco（西班牙安达卢西亚议会议员）。

C11 是一种强大的编程语言，它可以帮助你快速有效地编写高性能程序。你可以很好地在企业系统开发领域中使用 C11，来帮助他们建立业务关键型和任务关键型信息系统。我们真诚地欢迎读者提出可以提高本书质量的意见、批评、更正和建议，请将它们发送到：

deitel@deitel.com

本书的勘误放在：

www.deitel.com/books/cfp

我们希望你享受本书的阅读，就像我们编写本书时那样。

Paul Deitel

Harvey Deitel

作者简介

Paul Deitel，Deitel & Associates 有限公司的 CEO 兼 CTO，毕业于麻省理工学院，主修信息技术。在 Deitel & Associate 有限公司公司工作的过程中，他已经为行业、政府机关和军队客户提供了数百节编程课程，这些客户包括思科、IBM、西门子、Sun Microsystems、戴尔、Fidelity、肯尼迪航天中心、美国国家强风暴实验室、白沙导弹试验场、Rogue Wave Software、波音公司、SunGard Higher Education、北电网络公司、彪马、iRobot、Invensys 等。他和本书的合著者 Harvey M. Deitel 博士是全球畅销编程语言教材、专业书籍和视频的作者。

Harvey Deitel 博士，Deitel & Associates 有限公司的董事长和首席战略官，在计算机领域中拥有 50 多年的经验。Deitel 博士获得了麻省理工学院电子工程（学习计算）的学士和硕士学位，并获得了波士顿大学的数学博士学位（学习计算机科学）。他拥有丰富的行业 and 大学教学经验，在 1991 年与儿子 Paul Deitel 创办 Deitel & Associates 有限公司之前，他是波士顿大学计算机科学系的主任并获得了终身任职权。Deitel 博士为很多大公司、学术研究机构、政府机关和军方提供了数百场专业编程讲座。Deitel 的出版物获得了国际上的认可，并被翻译为繁体中文、简体中文、韩语、日语、德语、俄语、西班牙语、法语、波兰语、意大利语、葡萄牙语、希腊语、乌尔都语和土耳其语。

Deitel & Associates 有限公司的企业培训

由 Paul Deitel 和 Harvey Deitel 创立的 Deitel & Associates 有限公司是一家国际知名的写作、企业培训和软件开发公司，该公司专注于计算机编程语言、对象技术、Android 和 iOS 应用程序开发，以及 Internet 和 Web 软件技术方面的培训和写作。该公司向全球客户提供了由讲师主导的主要编程语言和平台课程，包括 C、C++、Visual C++、Java、Visual C#、Visual Basic、XML、Python、对象技术、互联网和 Web 编程、Android 应用程序开发、Objective-C 和 iOS 应用程序开发，并且还在不断提供其他编程语言和软件开发课程。公司的客户包括一些全球最大的公司、政府机关、军队和学术研究机构。

通过与 Prentice Hall/Pearson 37 年的合作，Deitel & Associates 有限公司出版了一流的专业编程书籍、大学教科书和 LiveLessons 视频课程。读者可以通过以下邮件地址联系 Deitel & Associates 有限公司和作者：

deitel@deitel.com

要了解更多 Deitel Dive-Into® 系列企业培训课程的信息，请访问：

www.deitel.com/training

如果贵机构希望获得关于由讲师主导的现场培训的相关建议，可发送邮件到 deitel@deitel.com。

只要订阅 Safari Books Online，就可以获取本书的电子版本，订阅地址为：

www.safaribooksonline.com

本书的最后一页会告诉你如何获得 45 天的免费试用期，这样你就可以访问本书的电子版本了。

个人想购买 Deitel 书籍和 LiveLessons 视频培训，可以访问 www.deitel.com。公司、政府机关、军队和学术研究组织的大宗团购应该直接与 Pearson 联系。要了解更多信息，请访问：

www.informit.com/store/sales.aspx

前言

xiii

I	Introduction	I
1.1	Introduction	2
1.2	The C Programming Language	2
1.3	C Standard Library	4
1.4	C++ and Other C-Based Languages	4
1.5	Typical C Program Development Environment	5
1.5.1	Phase 1: Creating a Program	6
1.5.2	Phases 2 and 3: Preprocessing and Compiling a C Program	7
1.5.3	Phase 4: Linking	7
1.5.4	Phase 5: Loading	7
1.5.5	Phase 6: Execution	7
1.5.6	Standard Input, Standard Output and Standard Error Streams	8
1.6	Test-Driving a C Application in Windows, Linux and Mac OS X	8
1.6.1	Running a C Application from the Windows Command Prompt	9
1.6.2	Running a C Application Using GNU C with Linux	11
1.6.3	Running a C Application Using GNU C with Mac OS X	14
1.7	Operating Systems	16
1.7.1	Windows—A Proprietary Operating System	17
1.7.2	Linux—An Open-Source Operating System	17
1.7.3	Apple's Mac OS X; Apple's iOS® for iPhone®, iPad® and iPod Touch® Devices	17
1.7.4	Google's Android	18
2	Introduction to C Programming	19
2.1	Introduction	20
2.2	A Simple C Program: Printing a Line of Text	20
2.3	Another Simple C Program: Adding Two Integers	24
2.4	Arithmetic in C	27
2.5	Decision Making: Equality and Relational Operators	31
2.6	Secure C Programming	35

3	Control Statements: Part I	37
3.1	Introduction	38
3.2	Control Structures	38
3.3	The if Selection Statement	40
3.4	The if...else Selection Statement	40
3.5	The while Repetition Statement	43
3.6	Class Average with Counter-Controlled Repetition	44
3.7	Class Average with Sentinel-Controlled Repetition	46
3.8	Nested Control Statements	49
3.9	Assignment Operators	51
3.10	Increment and Decrement Operators	52
3.11	Secure C Programming	55
4	Control Statements: Part II	57
4.1	Introduction	58
4.2	Repetition Essentials	58
4.3	Counter-Controlled Repetition	59
4.4	for Repetition Statement	60
4.5	for Statement: Notes and Observations	63
4.6	Examples Using the for Statement	64
4.7	switch Multiple-Selection Statement	67
4.8	do...while Repetition Statement	73
4.9	break and continue Statements	75
4.10	Logical Operators	77
4.11	Confusing Equality (==) and Assignment (=) Operators	80
4.12	Secure C Programming	81
5	Functions	83
5.1	Introduction	84
5.2	Program Modules in C	84
5.3	Math Library Functions	85
5.4	Functions	86
5.5	Function Definitions	87
5.6	Function Prototypes: A Deeper Look	91
5.7	Function Call Stack and Stack Frames	94
5.8	Headers	97
5.9	Passing Arguments By Value and By Reference	98
5.10	Random Number Generation	99
5.11	Example: A Game of Chance	104
5.12	Storage Classes	107
5.13	Scope Rules	109
5.14	Recursion	112
5.15	Example Using Recursion: Fibonacci Series	116

5.16	Recursion vs. Iteration	119
5.17	Secure C Programming	121

6 Arrays 122

6.1	Introduction	123
6.2	Arrays	123
6.3	Defining Arrays	124
6.4	Array Examples	125
6.5	Passing Arrays to Functions	138
6.6	Sorting Arrays	142
6.7	Case Study: Computing Mean, Median and Mode Using Arrays	144
6.8	Searching Arrays	149
6.9	Multidimensional Arrays	155
6.10	Variable-Length Arrays	162
6.11	Secure C Programming	165

7 Pointers 167

7.1	Introduction	168
7.2	Pointer Variable Definitions and Initialization	168
7.3	Pointer Operators	169
7.4	Passing Arguments to Functions by Reference	172
7.5	Using the const Qualifier with Pointers	176
7.5.1	Converting a String to Uppercase Using a Non-Constant Pointer to Non-Constant Data	177
7.5.2	Printing a String One Character at a Time Using a Non-Constant Pointer to Constant Data	178
7.5.3	Attempting to Modify a Constant Pointer to Non-Constant Data	180
7.5.4	Attempting to Modify a Constant Pointer to Constant Data	181
7.6	Bubble Sort Using Pass-by-Reference	182
7.7	sizeof Operator	185
7.8	Pointer Expressions and Pointer Arithmetic	188
7.9	Relationship between Pointers and Arrays	190
7.10	Arrays of Pointers	194
7.11	Case Study: Card Shuffling and Dealing Simulation	195
7.12	Pointers to Functions	199
7.13	Secure C Programming	204

8 Characters and Strings 205

8.1	Introduction	206
8.2	Fundamentals of Strings and Characters	206
8.3	Character-Handling Library	208
8.3.1	Functions isdigit, isalpha, isalnum and isxdigit	209
8.3.2	Functions islower, isupper, tolower and toupper	211
8.3.3	Functions isspace, iscntrl, ispunct, isprint and isgraph	212

8.4	String-Conversion Functions	213
8.4.1	Function strtod	214
8.4.2	Function strtol	215
8.4.3	Function strtoul	216
8.5	Standard Input/Output Library Functions	217
8.5.1	Functions fgets and putchar	217
8.5.2	Function getchar	219
8.5.3	Function sprintf	220
8.5.4	Function sscanf	220
8.6	String-Manipulation Functions of the String-Handling Library	221
8.6.1	Functions strcpy and strncpy	222
8.6.2	Functions strcat and strncat	223
8.7	Comparison Functions of the String-Handling Library	224
8.8	Search Functions of the String-Handling Library	225
8.8.1	Function strchr	226
8.8.2	Function strcspn	227
8.8.3	Function strpbrk	228
8.8.4	Function strrchr	228
8.8.5	Function strspn	229
8.8.6	Function strstr	229
8.8.7	Function strtok	230
8.9	Memory Functions of the String-Handling Library	231
8.9.1	Function memcpy	232
8.9.2	Function memmove	233
8.9.3	Function memcmp	234
8.9.4	Function memchr	234
8.9.5	Function memset	235
8.10	Other Functions of the String-Handling Library	236
8.10.1	Function strerror	236
8.10.2	Function strlen	236
8.11	Secure C Programming	237

9 Formatted Input/Output 238

9.1	Introduction	239
9.2	Streams	239
9.3	Formatting Output with printf	239
9.4	Printing Integers	240
9.5	Printing Floating-Point Numbers	241
9.6	Printing Strings and Characters	243
9.7	Other Conversion Specifiers	244
9.8	Printing with Field Widths and Precision	245
9.9	Using Flags in the printf Format Control String	247
9.10	Printing Literals and Escape Sequences	250
9.11	Reading Formatted Input with scanf	251
9.12	Secure C Programming	257

10 Structures, Unions, Bit Manipulation and Enumerations 258

10.1	Introduction	259
10.2	Structure Definitions	259
10.2.1	Self-Referential Structures	260
10.2.2	Defining Variables of Structure Types	260
10.2.3	Structure Tag Names	261
10.2.4	Operations That Can Be Performed on Structures	261
10.3	Initializing Structures	262
10.4	Accessing Structure Members	262
10.5	Using Structures with Functions	264
10.6	typedef	264
10.7	Example: High-Performance Card Shuffling and Dealing Simulation	265
10.8	Unions	268
10.8.1	Union Declarations	268
10.8.2	Operations That Can Be Performed on Unions	268
10.8.3	Initializing Unions in Declarations	269
10.8.4	Demonstrating Unions	269
10.9	Bitwise Operators	270
10.9.1	Displaying an Unsigned Integer in Bits	271
10.9.2	Making Function <code>displayBits</code> More Scalable and Portable	273
10.9.3	Using the Bitwise AND, Inclusive OR, Exclusive OR and Complement Operators	273
10.9.4	Using the Bitwise Left- and Right-Shift Operators	276
10.9.5	Bitwise Assignment Operators	278
10.10	Bit Fields	279
10.11	Enumeration Constants	282
10.12	Secure C Programming	284

11 File Processing 285

11.1	Introduction	286
11.2	Files and Streams	286
11.3	Creating a Sequential-Access File	287
11.4	Reading Data from a Sequential-Access File	292
11.5	Random-Access Files	296
11.6	Creating a Random-Access File	297
11.7	Writing Data Randomly to a Random-Access File	299
11.8	Reading Data from a Random-Access File	302
11.9	Case Study: Transaction-Processing Program	303
11.10	Secure C Programming	309

12 Data Structures 311

12.1	Introduction	312
12.2	Self-Referential Structures	312

12.3	Dynamic Memory Allocation	313
12.4	Linked Lists	314
12.4.1	Function insert	320
12.4.2	Function delete	321
12.4.3	Function printList	322
12.5	Stacks	323
12.5.1	Function push	327
12.5.2	Function pop	328
12.5.3	Applications of Stacks	328
12.6	Queues	329
12.6.1	Function enqueue	333
12.6.2	Function dequeue	334
12.7	Trees	335
12.7.1	Function insertNode	338
12.7.2	Traversals: Functions inOrder, preOrder and postOrder	339
12.7.3	Duplicate Elimination	340
12.7.4	Binary Tree Search	340
12.8	Secure C Programming	340

13 Preprocessor **342**

13.1	Introduction	343
13.2	#include Preprocessor Directive	343
13.3	#define Preprocessor Directive: Symbolic Constants	344
13.4	#define Preprocessor Directive: Macros	344
13.5	Conditional Compilation	346
13.6	#error and #pragma Preprocessor Directives	347
13.7	# and ## Operators	348
13.8	Line Numbers	348
13.9	Predefined Symbolic Constants	348
13.10	Assertions	349
13.11	Secure C Programming	349

14 Other Topics **351**

14.1	Introduction	352
14.2	Redirecting I/O	352
14.3	Variable-Length Argument Lists	353
14.4	Using Command-Line Arguments	355
14.5	Notes on Compiling Multiple-Source-File Programs	356
14.6	Program Termination with exit and atexit	358
14.7	Suffixes for Integer and Floating-Point Literals	359
14.8	Signal Handling	360
14.9	Dynamic Memory Allocation: Functions calloc and realloc	362
14.10	Unconditional Branching with goto	363

A	Operator Precedence Chart	365
B	ASCII Character Set	367
C	Number Systems	368
C.1	Introduction	369
C.2	Abbreviating Binary Numbers as Octal and Hexadecimal Numbers	372
C.3	Converting Octal and Hexadecimal Numbers to Binary Numbers	373
C.4	Converting from Binary, Octal or Hexadecimal to Decimal	373
C.5	Converting from Decimal to Binary, Octal or Hexadecimal	374
C.6	Negative Binary Numbers: Two's Complement Notation	376
D	Sorting: A Deeper Look	378
D.1	Introduction	379
D.2	Big O Notation	379
D.3	Selection Sort	380
D.4	Insertion Sort	384
D.5	Merge Sort	387
E	Additional Features of the C Standard	394
E.1	Introduction	395
E.2	Support for C99	396
E.3	C99 Headers	396
E.4	Mixing Declarations and Executable Code	397
E.5	Declaring a Variable in a for Statement Header	397
E.6	Designated Initializers and Compound Literals	398
E.7	Type <code>bool</code>	401
E.8	Implicit <code>int</code> in Function Declarations	402
E.9	Complex Numbers	403
E.10	Variable-Length Arrays	404
E.11	Additions to the Preprocessor	407
E.12	Other C99 Features	408
E.12.1	Compiler Minimum Resource Limits	408
E.12.2	The <code>restrict</code> Keyword	409
E.12.3	Reliable Integer Division	409
E.12.4	Flexible Array Members	409
E.12.5	Relaxed Constraints on Aggregate Initialization	410
E.12.6	Type Generic Math	410
E.12.7	Inline Functions	410
E.12.8	<code>return</code> Without Expression	411
E.12.9	<code>__func__</code> Predefined Identifier	411
E.12.10	<code>va_copy</code> Macro	411

E.13	New Features in the C11 Standard	411
E.13.1	New C11 Headers	412
E.13.2	Multithreading Support	412
E.13.3	quick_exit function	420
E.13.4	Unicode® Support	420
E.13.5	_Noreturn Function Specifier	420
E.13.6	Type-Generic Expressions	420
E.13.7	Annex L: Analyzability and Undefined Behavior	421
E.13.8	Anonymous Structures and Unions	421
E.13.9	Memory Alignment Control	422
E.13.10	Static Assertions	422
E.13.11	Floating Point Types	422
E.14	Web Resources	422

F Using the Visual Studio Debugger 425

F.1	Introduction	426
F.2	Breakpoints and the Continue Command	426
F.3	Locals and Watch Windows	430
F.4	Controlling Execution Using the Step Into, Step Over, Step Out and Continue Commands	432
F.5	Autos Window	434

G Using the GNU Debugger 436

G.1	Introduction	437
G.2	Breakpoints and the run, stop, continue and print Commands	437
G.3	print and set Commands	442
G.4	Controlling Execution Using the step, finish and next Commands	444
G.5	watch Command	446

Index 449