

Why Does $2+2=5986$?

3rd Edition



Practical C

Programming

O'REILLY®

Steve Oualline

Practical C Programming



There are lots of introductory C books, but this is the first one that has the no-nonsense, practical approach that has made Nutshell Handbooks famous.

C programming is more than just getting the syntax right. Style and debugging also play a tremendous part in creating programs that run well and are easy to maintain. This book not only teaches you the mechanics of programming, but also describes how to create programs that are easy to read, debug, and update.

Practical rules are stressed. For example, there are fifteen precedence rules in C (&& comes before || comes before ?:). The practical programmer reduces these to two:

- Multiplication and division come before addition and subtraction.
- Put parentheses around everything else.

Contrary to popular belief, most programmers do not spend their time creating code. Most of their time is spent modifying someone else's code. This book shows you how to avoid the all-too-common obfuscated uses of C and also to recognize these uses when you encounter them in existing programs. You learn to leave code that other programmers do not have to struggle with. Electronic archaeology, the art of going through someone else's code, is described.

This third edition introduces popular Integrated Development Environments on Windows systems, as well as Unix programming utilities, and features a large statistics-generating program to pull together the concepts and features in the language.

oreilly.com

US \$34.95

CAN \$51.95

ISBN: 978-1-56592-306-5



Practical C Programming

Practical C Programming

Third Edition

Steve Oualline

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

Practical C Programming, Third Edition

by Steve Oualline

Copyright © 1997, 1993, 1991 O'Reilly Media, Inc. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Editor, First and Second Editions: Tim O'Reilly

Editor, Third Edition: Andy Oram

Production Editor: Nicole Gipson Arigo

Printing History:

July 1991:	First Edition.
August 1992:	Minor corrections.
January 1993:	Second Edition. Bugs fixed; programs all now conform to ANSI C.
June 1993:	Minor corrections.
August 1997:	Third Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks and The Java Series is a trademark of O'Reilly Media, Inc. *Practical C Programming, Third Edition*, the image of a cow, and related trade dress are trademarks of O'Reilly Media, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Table of Contents

<i>Preface</i>	<i>xv</i>
How This Book is Organized	xvi
Chapter by Chapter	xvi
Notes on the Third Edition	xix
Font Conventions	xix
Obtaining Source Code	xx
Comments and Questions	xxii
Acknowledgments	xxii
Acknowledgments to the Third Edition	xxiii
 <i>I. Basics</i>	 <i>1</i>
 <i>1. What Is C?</i>	 <i>3</i>
How Programming Works	4
Brief History of C	6
How C Works	7
How to Learn C	9
 <i>2. Basics of Program Writing</i>	 <i>11</i>
Programs from Conception to Execution	11
Creating a Real Program	12
Creating a Program Using a Command-Line Compiler	13
Creating a Program Using an Integrated Development Environment	16
Getting Help on UNIX	33

Getting Help in an Integrated Development Environment	34
IDE Cookbooks	34
Programming Exercises	37
3. <i>Style</i>	38
Common Coding Practices	42
Coding Religion	45
Indentation and Code Format	45
Clarity	46
Simplicity	47
Summary	48
4. <i>Basic Declarations and Expressions</i>.....	49
Elements of a Program	49
Basic Program Structure	50
Simple Expressions	51
Variables and Storage	52
Variable Declarations	53
Integers	54
Assignment Statements	54
printf Function	56
Floating Point	57
Floating Point Versus Integer Divide	58
Characters	60
Answers	61
Programming Exercises	62
5. <i>Arrays, Qualifiers, and Reading Numbers</i>	63
Arrays	63
Strings	64
Reading Strings	67
Multidimensional Arrays	70
Reading Numbers	71
Initializing Variables	73
Types of Integers	75
Types of Floats	77
Constant Declarations	77

Hexadecimal and Octal Constants	78
Operators for Performing Shortcuts	78
Side Effects	79
++x or x++	80
More Side-Effect Problems	81
Answers	82
Programming Exercises	83
 6. <i>Decision and Control Statements</i>	84
if Statement	84
else Statement	85
How Not to Use strcmp	86
Looping Statements	87
while Statement	87
break Statement	89
continue Statement	90
Assignment Anywhere Side Effect	91
Answer	92
Programming Exercises	93
 7. <i>Programming Process</i>	95
Setting Up	97
Specification	98
Code Design	98
Prototype	99
Makefile	101
Testing	104
Debugging	104
Maintenance	107
Revisions	107
Electronic Archaeology	107
Marking Up the Program	108
Using the Debugger	108
Text Editor as a Browser	108
Add Comments	109
Programming Exercises	111

<i>II. Simple Programming</i>	<i>113</i>
8. <i>More Control Statements</i>	<i>115</i>
for Statement	115
switch Statement	118
switch, break, and continue	123
Answers	124
Programming Exercises	125
9. <i>Variable Scope and Functions</i>	<i>127</i>
Scope and Class	127
Functions	130
Functions with No Parameters	134
Structured Programming	135
Recursion	136
Answers	138
Programming Exercises	138
10. <i>C Preprocessor</i>	<i>140</i>
#define Statement	140
Conditional Compilation	146
include Files	148
Parameterized Macros	150
Advanced Features	152
Summary	152
Answers	152
Programming Exercises	154
11. <i>Bit Operations</i>	<i>156</i>
Bit Operators	158
The and Operator (&)	158
Bitwise or ()	160
The Bitwise Exclusive or (^)	161
The Ones Complement Operator (Not) (~)	161
The Left- and Right-Shift Operators (<<, >>)	162
Setting, Clearing, and Testing Bits	163
Bitmapped Graphics	166

