

Contents

About the Author	xix
Acknowledgments	xxi
Introduction	xxiii

CHAPTER 1 Programming in C	1
Creating C Programs	1
Editing	1
Compiling	2
Linking	3
Executing	3
Creating Your First Program	4
Editing Your First Program	5
Dealing with Errors	6
Dissecting a Simple Program	7
Comments	7
Preprocessing Directives	8
Defining the main() Function	9
Keywords	9
The Body of a Function	10
Outputting Information	10
Arguments	11
Control Characters	11
Developing Programs in C	13
Understanding the Problem	13
Detailed Design	14
Implementation	14
Testing	14
Functions and Modular Programming	14
Common Mistakes	18
Points to Remember	18
Summary	19
Exercises	20

CHAPTER 2	First Steps in Programming	21
	Memory in Your Computer	21
	What Is a Variable?	23
	Variables That Store Numbers	24
	Integer Variables	24
	Naming Variables	27
	Using Variables	29
	Initializing Variables	30
	Arithmetic Statements	31
	Variables and Memory	37
	Integer Variable Types	38
	Unsigned Integer Types	38
	Using Integer Types	39
	Specifying Integer Constants	40
	Floating-Point Values	41
	Floating-Point Variables	42
	Division Using Floating-Point Values	43
	Controlling the Number of Decimal Places	44
	Controlling the Output Field Width	44
	More Complicated Expressions	45
	Defining Constants	48
	Knowing Your Limitations	50
	Introducing the sizeof Operator	52
	Choosing the Correct Type for the Job	54
	Explicit Type Conversion	57
	Automatic Conversion	57
	Rules for Implicit Conversions	57
	Implicit Conversions in Assignment Statements	58
	More Numeric Data Types	59
	The Character Type	59
	Character Input and Character Output	60
	The Wide Character Type	63
	Enumerations	64
	Variables to Store Boolean Values	66
	The Complex Number Types	67

The op= Form of Assignment	70
Mathematical Functions	71
Designing a Program	72
The Problem	72
The Analysis	73
The Solution	75
Summary	79
Exercises	80

CHAPTER 3 Making Decisions

The Decision-Making Process	81
Arithmetic Comparisons	82
Expressions Involving Relational Operators	82
The Basic if Statement	82
Extending the if Statement: if-else	86
Using Blocks of Code in if Statements	88
Nested if Statements	89
More Relational Operators	92
Logical Operators	96
The Conditional Operator	99
Operator Precedence: Who Goes First?	102
Multiple-Choice Questions	106
Using else-if Statements for Multiple Choices	106
The switch Statement	107
The goto Statement	115
Bitwise Operators	116
The op= Use of Bitwise Operators	119
Using Bitwise Operators	119
Designing a Program	122
The Problem	122
The Analysis	122
The Solution	123
Summary	126
Exercises	126

CHAPTER 4	Loops	129
	How Loops Work	129
	Introducing the Increment and Decrement Operators	130
	The for Loop	131
	General Syntax of the for Loop	135
	More on the Increment and Decrement Operators	136
	The Increment Operator	136
	The Prefix and Postfix Forms of the Increment Operator	137
	The Decrement Operator	137
	The for Loop Revisited	138
	Modifying the for Loop Variable	140
	A for Loop with No Parameters	141
	The break Statement in a Loop	141
	Limiting Input Using a for Loop	144
	Generating Pseudo-Random Integers	146
	More for Loop Control Options	148
	Floating-Point Loop Control Variables	149
	The while Loop	149
	Nested Loops	153
	Nested Loops and the goto Statement	156
	The do-while Loop	157
	The continue Statement	160
	Designing a Program	160
	The Problem	160
	The Analysis	160
	The Solution	162
	Summary	172
	Exercises	173
CHAPTER 5	Arrays	175
	An Introduction to Arrays	175
	Programming Without Arrays	175
	What Is an Array?	177
	Using Arrays	178
	A Reminder About Memory	181

Arrays and Addresses	184
Initializing an Array	186
Finding the Size of an Array	186
Multidimensional Arrays	187
Initializing Multidimensional Arrays	189
Designing a Program	194
The Problem	194
The Analysis	194
The Solution	195
Summary	202
Exercises	202

CHAPTER 6 Applications with Strings and Text 203

What Is a String?	203
String- and Text-Handling Methods	205
Operations with Strings	208
Appending a String	208
Arrays of Strings	210
String Library Functions	212
Copying Strings Using a Library Function	212
Determining String Length Using a Library Function	213
Joining Strings Using a Library Function	214
Comparing Strings	215
Searching a String	218
Analyzing and Transforming Strings	221
Converting Characters	224
Converting Strings to Numerical Values	227
Working with Wide Character Strings	227
Operations on Wide Character Strings	228
Testing and Converting Wide Characters	229
Designing a Program	231
The Problem	231
The Analysis	231
The Solution	231
Summary	238
Exercises	239

CHAPTER 7	Pointers	241
	A First Look at Pointers	241
	Declaring Pointers	242
	Accessing a Value Through a Pointer	243
	Using Pointers	246
	Pointers to Constants	250
	Constant Pointers	251
	Naming Pointers	251
	Arrays and Pointers	251
	Multidimensional Arrays	255
	Multidimensional Arrays and Pointers	259
	Accessing Array Elements	260
	Using Memory As You Go	263
	Dynamic Memory Allocation: The malloc() Function	263
	Memory Allocation with the calloc() Function	268
	Releasing Dynamically Allocated Memory	268
	Reallocating Memory	270
	Handling Strings Using Pointers	271
	String Input with More Control	272
	Using Arrays of Pointers	273
	Designing a Program	283
	The Problem	283
	The Analysis	284
	The Solution	284
	Summary	294
	Exercises	294
CHAPTER 8	Structuring Your Programs	295
	Program Structure	295
	Variable Scope and Lifetime	296
	Variable Scope and Functions	299
	Functions	299
	Defining a Function	300
	The return Statement	304

The Pass-By-Value Mechanism	307
Function Declarations	309
Pointers As Arguments and Return Values	310
const Parameters	313
Returning Pointer Values from a Function	322
Incrementing Pointers in a Function	326
Summary	326
Exercises	327

CHAPTER 9 More on Functions	329
Pointers to Functions	329
Declaring a Pointer to a Function	329
Calling a Function Through a Function Pointer	330
Arrays of Pointers to Functions	333
Pointers to Functions As Arguments	335
Variables in Functions	338
Static Variables: Keeping Track Within a Function	338
Sharing Variables Between Functions	340
Functions That Call Themselves: Recursion	343
Functions with a Variable Number of Arguments	345
Copying a va_list	348
Basic Rules for Variable-Length Argument Lists	348
The main() Function	349
Ending a Program	350
Libraries of Functions: Header Files	351
Enhancing Performance	352
Declaring Functions inline	352
Using the restrict Keyword	353
Designing a Program	353
The Problem	353
The Analysis	354
The Solution	356
Summary	371
Exercises	372

CHAPTER 10	Essential Input and Output Operations	373
	Input and Output Streams	373
	Standard Streams	374
	Input from the Keyboard	375
	Formatted Keyboard Input	376
	Input Format Control Strings	376
	Characters in the Input Format String	382
	Variations on Floating-Point Input	383
	Reading Hexadecimal and Octal Values	384
	Reading Characters Using <code>scanf()</code>	386
	Pitfalls with <code>scanf()</code>	388
	String Input from the Keyboard	388
	Unformatted Input from the Keyboard	389
	Output to the Screen	394
	Formatted Output to the Screen Using <code>printf()</code>	394
	Escape Sequences	396
	Integer Output	397
	Outputting Floating-Point Values	400
	Character Output	401
	Other Output Functions	403
	Unformatted Output to the Screen	404
	Formatted Output to an Array	404
	Formatted Input from an Array	405
	Sending Output to the Printer	405
	Summary	406
	Exercises	406
CHAPTER 11	Structuring Data	409
	Data Structures: Using <code>struct</code>	409
	Defining Structure Types and Structure Variables	411
	Accessing Structure Members	411
	Unnamed Structures	414
	Arrays of Structures	414
	Structures in Expressions	417
	Pointers to Structures	417
	Dynamic Memory Allocation for Structures	418

More on Structure Members	420
Structures As Members of a Structure	420
Declaring a Structure Within a Structure	421
Pointers to Structures As Structure Members	422
Doubly Linked Lists	426
Bit-Fields in a Structure	429
Structures and Functions	430
Structures As Arguments to Functions	430
Pointers to Structures As Function Arguments	431
A Structure As a Function Return Value	432
An Exercise in Program Modification	436
Binary Trees	439
Sharing Memory	447
Unions	448
Pointers to Unions	450
Initializing Unions	450
Structures As Union Members	450
Defining Your Own Data Types	451
Structures and the typedef Facility	452
Simplifying Code Using typedef	453
Designing a Program	454
The Problem	454
The Analysis	454
The Solution	454
Summary	464
Exercises	465

■ CHAPTER 12 Working with Files

The Concept of a File	467
Positions in a File	468
File Streams	468
Accessing Files	468
Opening a File	469
Renaming a File	471
Closing a File	472
Deleting a File	472

Writing to a Text File	473
Reading from a Text File	474
Writing Strings to a Text File	476
Reading Strings from a Text File	477
Formatted File Input and Output	480
Formatted Output to a File	481
Formatted Input from a File	481
Dealing with Errors	483
Further Text File Operation Modes	484
Binary File Input and Output	485
Specifying Binary Mode	486
Writing a Binary File	486
Reading a Binary File	487
Moving Around in a File	495
File Positioning Operations	495
Finding Out Where You Are	496
Setting a Position in a File	496
Using Temporary Work Files	502
Creating a Temporary Work File	502
Creating a Unique File Name	502
Updating Binary Files	503
Changing the File Contents	508
Reading a Record from the Keyboard	509
Writing a Record to a File	510
Reading a Record from a File	511
Writing a File	512
Listing the File Contents	513
Updating the Existing File Contents	514
File Open Modes Summary	521
Designing a Program	521
The Problem	522
The Analysis	522
The Solution	522
Summary	527
Exercises	527

CHAPTER 13	Supporting Facilities	529
	Preprocessing	529
	Including Header Files in Your Programs	530
	External Variables and Functions	530
	Substitutions in Your Program Source Code	531
	Macro Substitutions	532
	Macros That Look Like Functions	532
	Preprocessor Directives on Multiple Lines	534
	Strings As Macro Arguments	534
	Joining Two Results of a Macro Expansion	535
	Logical Preprocessor Directives	536
	Conditional Compilation	536
	Directives Testing for Specific Values	537
	Multiple-Choice Selections	537
	Standard Preprocessing Macros	538
	Debugging Methods	538
	Integrated Debuggers	539
	The Preprocessor in Debugging	539
	Using the assert() Macro	543
	Additional Library Functions	545
	The Date and Time Function Library	545
	Getting the Date	549
	Summary	555
	Exercises	555
APPENDIX A	Computer Arithmetic	557
	Binary Numbers	557
	Hexadecimal Numbers	558
	Negative Binary Numbers	560
	Big-Endian and Little-Endian Systems	561
	Floating-Point Numbers	562
APPENDIX B	ASCII Character Code Definitions	565
APPENDIX C	Reserved Words in C	571

APPENDIX D

Input and Output Format Specifications

573

Output Format Specifications

573

Input Format Specifications

576

INDEX

579