

Table of Contents

Introduction 1

1 Some Fundamentals 5

- Programming 5
- Higher-Level Languages 5
- Operating Systems 6
- Compiling Programs 7
- Integrated Development Environments 10
- Language Interpreters 10

2 Compiling and Running Your First Program 11

- Compiling Your Program 12
- Running Your Program 12
- Understanding Your First Program 13
- Displaying the Values of Variables 15
- Comments 17
- Exercises 19

3 Variables, Data Types, and Arithmetic Expressions 21

- Understanding Data Types and Constants 21
 - The Integer Type `int` 22
 - The Floating Number Type `float` 23
 - The Extended Precision Type `double` 23
 - The Single Character Type `char` 24
 - The Boolean Data Type `_Bool` 24
 - Type Specifiers: `long`, `long long`, `short`, `unsigned`, and `signed` 26
- Working with Variables 29
- Working with Arithmetic Expressions 30
 - Integer Arithmetic and the Unary Minus Operator 33
- Combining Operations with Assignment: The Assignment Operators 39
- Types `_Complex` and `_Imaginary` 40
- Exercises 40

4 Program Looping 43

- Triangular Numbers 43
- The `for` Statement 44
 - Relational Operators 46
 - Aligning Output 50
- Program Input 51
 - Nested `for` Loops 53
 - `for` Loop Variants 55
- The `while` Statement 56
- The `do` Statement 60
 - The `break` Statement 62
 - The `continue` Statement 62
- Exercises 63

5 Making Decisions 65

- The `if` Statement 65
 - The `if-else` Construct 69
 - Compound Relational Tests 72
 - Nested `if` Statements 74
 - The `else if` Construct 76
- The `switch` Statement 83
- Boolean Variables 86
- The Conditional Operator 90
- Exercises 92

6 Working with Arrays 95

- Defining an Array 96
 - Using Array Elements as Counters 100
 - Generating Fibonacci Numbers 103
 - Using an Array to Generate Prime Numbers 104
- Initializing Arrays 106
- Character Arrays 108
 - Base Conversion Using Arrays 109
 - The `const` Qualifier 111
- Multidimensional Arrays 113
- Variable Length Arrays 115
- Exercises 117

- 7 Working with Functions 119**
 - Defining a Function 119
 - Arguments and Local Variables 123
 - Function Prototype Declaration 124
 - Automatic Local Variables 124
 - Returning Function Results 126
 - Functions Calling Functions Calling... 130
 - Declaring Return Types and Argument Types 133
 - Checking Function Arguments 135
 - Top-Down Programming 137
 - Functions and Arrays 137
 - Assignment Operators 141
 - Sorting Arrays 143
 - Multidimensional Arrays 146
 - Global Variables 151
 - Automatic and Static Variables 155
 - Recursive Functions 158
 - Exercises 161

- 8 Working with Structures 163**
 - The Basics of Structures 163
 - A Structure for Storing the Date 164
 - Using Structures in Expressions 166
 - Functions and Structures 169
 - A Structure for Storing the Time 175
 - Initializing Structures 178
 - Compound Literals 178
 - Arrays of Structures 180
 - Structures Containing Structures 183
 - Structures Containing Arrays 185
 - Structure Variants 189
 - Exercises 190

- 9 Character Strings 193**
 - Revisiting the Basics of Strings 193
 - Arrays of Characters 194

Variable-Length Character Strings	197
Initializing and Displaying Character Strings	199
Testing Two Character Strings for Equality	202
Inputting Character Strings	204
Single-Character Input	206
The Null String	211
Escape Characters	215
More on Constant Strings	217
Character Strings, Structures, and Arrays	218
A Better Search Method	221
Character Operations	226
Exercises	229
10 Pointers	233
Pointers and Indirection	233
Defining a Pointer Variable	234
Using Pointers in Expressions	237
Working with Pointers and Structures	239
Structures Containing Pointers	241
Linked Lists	243
The Keyword <code>const</code> and Pointers	251
Pointers and Functions	252
Pointers and Arrays	258
A Slight Digression About Program Optimization	262
Is It an Array or Is It a Pointer?	262
Pointers to Character Strings	264
Constant Character Strings and Pointers	266
The Increment and Decrement Operators Revisited	267
Operations on Pointers	271
Pointers to Functions	272
Pointers and Memory Addresses	273
Exercises	275
11 Operations on Bits	277
The Basics of Bits	277
Bit Operators	278
The Bitwise AND Operator	279

- The Bitwise Inclusive-OR Operator 281
- The Bitwise Exclusive-OR Operator 282
- The Ones Complement Operator 283
- The Left Shift Operator 285
- The Right Shift Operator 286
- A Shift Function 286
- Rotating Bits 288
- Bit Fields 291
- Exercises 295

12 The Preprocessor 297

- The `#define` Statement 297
 - Program Extendability 301
 - Program Portability 302
 - More Advanced Types of Definitions 304
 - The `#` Operator 309
 - The `##` Operator 310
- The `#include` Statement 311
 - System Include Files 313
- Conditional Compilation 314
 - The `#ifdef`, `#endif`, `#else`, and `#ifndef` Statements 314
 - The `#if` and `#elif` Preprocessor Statements 316
 - The `#undef` Statement 317
- Exercises 318

13 Extending Data Types with the Enumerated Data Type, Type Definitions, and Data Type Conversions 319

- Enumerated Data Types 319
- The `typedef` Statement 323
- Data Type Conversions 325
 - Sign Extension 327
 - Argument Conversion 328
- Exercises 329

14 Working with Larger Programs 331

- Dividing Your Program into Multiple Files 331
 - Compiling Multiple Source Files from the Command Line 332

Communication Between Modules	334
External Variables	334
Static Versus Extern Variables and Functions	337
Using Header Files Effectively	339
Other Utilities for Working with Larger Programs	341
The <code>make</code> Utility	341
The <code>cvs</code> Utility	343
Unix Utilities: <code>ar</code> , <code>grep</code> , <code>sed</code> , and so on	343
15 Input and Output Operations in C	345
Character I/O: <code>getchar()</code> and <code>putchar()</code>	346
Formatted I/O: <code>printf()</code> and <code>scanf()</code>	346
The <code>printf()</code> Function	346
The <code>scanf()</code> Function	353
Input and Output Operations with Files	358
Redirecting I/O to a File	358
End of File	361
Special Functions for Working with Files	362
The <code>fopen</code> Function	362
The <code>getc()</code> and <code>putc()</code> Functions	364
The <code>fclose()</code> Function	365
The <code>feof</code> Function	367
The <code>fprintf()</code> and <code>fscanf()</code> Functions	367
The <code>fgets()</code> and <code>fputs()</code> Functions	367
<code>stdin</code> , <code>stdout</code> , and <code>stderr</code>	368
The <code>exit()</code> Function	369
Renaming and Removing Files	370
Exercises	371
16 Miscellaneous and Advanced Features	373
Miscellaneous Language Statements	373
The <code>goto</code> Statement	373
The <code>null</code> Statement	374
Working with Unions	375
The Comma Operator	378
Type Qualifiers	379
The <code>register</code> Qualifier	379

The <code>volatile</code> Qualifier	379
The <code>restrict</code> Qualifier	379
Command-line Arguments	380
Dynamic Memory Allocation	384
The <code>calloc()</code> and <code>malloc()</code> Functions	385
The <code>sizeof</code> Operator	385
The <code>free</code> Function	387
Exercises	389
17 Debugging Programs	391
Debugging with the Preprocessor	391
Debugging Programs with <code>gdb</code>	397
Working with Variables	400
Source File Display	401
Controlling Program Execution	402
Getting a Stack Trace	406
Calling Functions and Setting Arrays and Structures	407
Getting Help with <code>gdb</code> Commands	408
Odds and Ends	410
18 Object-Oriented Programming	413
What Is an Object Anyway?	413
Instances and Methods	414
Writing a C Program to Work with Fractions	416
Defining an Objective-C Class to Work with Fractions	417
Defining a C++ Class to Work with Fractions	421
Defining a C# Class to Work with Fractions	424
A C Language Summary	427
1.0 Digraphs and Identifiers	427
2.0 Comments	429
3.0 Constants	429
4.0 Data Types and Declarations	432
5.0 Expressions	442
6.0 Storage Classes and Scope	456
7.0 Functions	458
8.0 Statements	460
9.0 The Preprocessor	464

B	The Standard C Library	471
	Standard Header Files	471
	String Functions	474
	Memory Functions	475
	Character Functions	476
	I/O Functions	477
	In-Memory Format Conversion Functions	482
	String-to-Number Conversion	483
	Dynamic Memory Allocation Functions	484
	Math Functions	485
	General Utility Functions	493
C	Compiling Programs with gcc	495
	General Command Format	495
	Command-Line Options	496
D	Common Programming Mistakes	499
E	Resources	505
	The C Programming Language	505
	C Compilers and Integrated Development Environments	506
	Miscellaneous	507
	Index	509