Chapter 1: Style and Program Organization 1

Program Organization 1

Automatic Generation of Program Documentation 3

Module Design 4

Libraries and Other Module Groupings 4

Multiple header files 5
One header does all 5
Mixed approach 6

Program Aesthetics 6

Code paragraphs 6
Statements 8

Chapter 2: File Basics, Comments, and Program Headings11

File Basics 11

The Comment 13

Graphics 15
Packing bits 16

Letting the Editor Help You 17

Beginning Comment Block 18

The sections of a heading 18
Other sections 21

Module Headings 21

Function Headings 22

When to Write Comments 23

Some Comments on Comments 24

Chapter 3: Variable Names 25

A Brief History of the Variable 25

Capitalization 27

Names You Must Never Use 28

Other Names Not To Use 28

Avoid Similar Names 29

Consistency in Naming 29

Which Word First 29

Standard Prefixes and Suffixes 30

Module Prefixes 31

Special Prefixes and Suffixes 31

When You Can Use Short Names. 32

argv, argc. 32

Microsoft Notation 32

Imaginative Solutions 34

Case studies. 35

- The C runtime library 35
- The UNIX kernel 36
- The Microsoft library 37
- The X Windows System 37

Variable Declaration Comments. 38

- Units 38
- Structures and unions 39
- Long declarations and comments 40
- Group similar declarations 41

Hidden Variables 42

Portable Types 42

Numbers 43

- Floating-point numbers. 43
- Hex numbers 44
- Long integers 44

Chapter 4: Statement Formatting 46

Formatting the Body of the Program 46

- Simplifying complex statements. 48
- Splitting long statements. 48
  - Splitting and parentheses. 49
  - Splitting a for statement. 50

    - Splitting a switch statement. 51
    - Conditional operators (? :). 52
  - Side effects 52
  - Assignments in other statements. 55
  - When to use two statements per line. 56
- Logic and Indentation 56
  - Indentation styles 58
    - Short form 58
    - Braces stand alone. 59
    - Braces indented too. 60
    - Variations.. 60
  - How much to indent 61
    - Two Spaces: 62
    - Four Spaces: 62
    - Eight Spaces:. 62

Chapter 5: Statement Details. 66

- Doing Nothing 66
- Arithmetic Statements. 66
- Function Headings 67
  - K&R style parameters. 67
  - Return type 68
- Number of parameters. 69
- Passing parameters in globals 72
  - XView style parameter passing 73
- The if Statement 74
  - if/else chains 75
  - if and the comma operator 76
- The while Statement 76
- The do/while Statement 77
- The for Statement 77
  - Missing parts of for loops 77
  - Overstuffed for loops 78
- The printf Statement 79
- goto and Labels 80

The switch Statement 80

Debug Printing 84

Shut up Statements 84

Chapter 6: Preprocessor. 87

Simple Define Statements. 87

Constant expressions. 88
#define constants vs. consts 88
#define vs. typedef 89
Abuse of #define directives 90
Keywords and standard functions. 90

Parameterized Macros. 91

Multi-line Macros 92

Macros and Subroutines 94

The #include Directive 95

Style for #Includes 95
Protecting against double #Includes 96

Conditional Compilation. 96

Where to define the control symbols 98
Commenting out code. 99

Chapter 7: Directory Organization and Makefile Style . . 102

Organizing Your Directories. 102

The make Program 103

Heading Comments. 103

Customization Information 104
Standard targets. 104
Macro Definitions 105
Common macro definitions. 106

Configurable variables 106

Major Targets 107

Other Targets 107

Special Rules. 107
Dependencies 108
Example 108
Common Expressions 110
Complexity 111
Portability Considerations. 112
Generic Makefiles. 112
Conclusion 113
Chapter 8: User-Friendly Programming 114
What Does User-Friendly Mean? 114
Law of Least Astonishment 114
Modeling the User. 115
Error Messages 115
The Command Interface 116
Help. 117
Safety Nets 117
Accelerators 118
Chapter 9: Rules 119