

The Elements
of
C++ *Style*

Trevor Misfeldt

CenterSpace Software

Gregory Bumgardner

Freelance Consultant

Andrew Gray

IntelliChem Inc.



CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS

The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

<http://www.cambridge.org>

© Cambridge University Press 2004

This book is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without
the written permission of Cambridge University Press.

First published 2004

Printed in the United States of America

Typefaces Adobe Garamond 10.5/12 pt. and ITC Avant Garde
System L^AT_EX 2_ε [TB]

A catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

The elements of C++ style / Trevor Misfeldt [et al.].

p. cm.

Includes bibliographical references and index.

ISBN 0-521-89308-9 (pbk.)

1. C++ (Computer program language) I. Misfeldt, Trevor, 1969–

QA76.73.C153E28 2004

005.13'3 – dc22

2003061316

ISBN 0 521 89308 9 paperback

Contents

Preface	vii
Audience	vii
1. Introduction	1
Disclaimer	2
Acknowledgments	2
2. General Principles	4
3. Formatting Conventions	7
3.1 Indentation	7
4. Naming Conventions	17
4.1 Preprocessor Macro Names	17
4.2 Type and Constant Names	18
4.3 Function Names	19
4.4 Variable and Parameter Names	21
4.5 General	25
5. Documentation Conventions	29
6. Programming Principles	41
6.1 Engineering	41
6.2 Class Design	45
6.3 Thread Safety and Concurrency	53

7. Programming Conventions	58
7.1 Preprocessor	58
7.2 Declarations	63
7.3 Scoping	67
7.4 Functions and Methods	68
7.5 Classes	74
7.6 Class Members	78
7.7 Operators	93
7.8 Templates	100
7.9 Type Safety, Casting, and Conversion	102
7.10 Initialization and Construction	112
7.11 Statements and Expressions	119
7.12 Control Flow	124
7.13 Error and Exception Handling	129
7.14 Efficiency	137
8. Packaging Conventions	141
8.1 Scoping	141
8.2 Organization	143
8.3 Files	147
Summary	151
Glossary	161
Bibliography	171
Index	173

1.

Introduction

style: 1b. the shadow-producing pin of a sundial.

2c. -the custom or plan followed in spelling, capitalization, punctuation, and typographic arrangement and display.

—*Webster's New Collegiate Dictionary*

The syntax of a programming language tells you what code it is possible to write—what machines will understand. Style tells you what you ought to write—what humans reading the code will understand. Code written with a consistent, simple style is maintainable, robust, and contains fewer bugs. Code written with no regard to style contains more bugs, and may simply be thrown away and rewritten rather than maintained.

Attending to style is particularly important when developing as a team. Consistent style facilitates communication, because it enables team members to read and understand each other's work more easily. In our experience, the value of consistent programming style grows exponentially with the number of people working with the code.

Our favorite style guides are classics: Strunk and White's *The Elements of Style*⁴ and Kernighan and Plauger's *The Elements of Programming Style*.⁵ These small books work because they

⁴ William Strunk, Jr., and E. B. White. *The Elements of Style, Fourth Edition*. (Allyn & Bacon, 2000).

⁵ Brian Kernighan, and P. J. Plauger. *The Elements of Programming Style*. (New York: McGraw-Hill, 1988).

2 THE ELEMENTS OF C++ STYLE

are simple: a list of rules, each containing a brief explanation and examples of correct, and sometimes incorrect, use. We followed the same pattern in this book. This simple treatment—a series of rules—enabled us to keep this book short and easy to understand.

Some of the advice that you read here may seem obvious to you, particularly if you've been writing code for a long time. Others may disagree with some of our specific suggestions about formatting or indentation. What we've tried to do here is distill many decades of experience into an easily accessible set of heuristics that encourage consistent coding practice (and hopefully help you avoid some C++ traps along the way). The idea is to provide a clear standard to follow so programmers can spend their time on solving the problems of their customers instead of worrying about things like naming conventions and formatting.

Disclaimer

We have dramatically simplified the code samples used in this book to highlight the concepts related to a particular rule. In many cases, these code fragments do not conform to conventions described elsewhere in this book—they lack real documentation and fail to meet certain minimum declarative requirements. Do not treat these fragments as definitive examples of real code!

Acknowledgments

Books like these are necessarily a team effort. Major contributions came from the original authors of *The Elements of Java Style*: Al Vermeulen, Scott Ambler, Greg Bumgardner, Eldon Metz, Trevor Misfeldt, Jim Shur, and Patrick Thompson. Both that book and this one have some roots in “C++

Design, Implementation, and Style Guide,” written by Tom Keffer, and the “Rogue Wave Java Style Guide,” and the “Ambysoft Inc. Coding Standards for Java,” documents to which Jeremy Smith, Tom Keffer, Wayne Gramlich, Pete Handsman, and Cris Perdue all contributed. We’d also like to thank our former colleagues at Rogue Wave Software, from whom we’ve learned a lot over the years, and who have thus contributed to this work in both tangible and intangible ways.

Thanks also to the reviewers who provided valuable feedback on drafts of this book, including Ken Baldwin, Brand Hunt, Jim Shur, and Steve Sneller.

This book would certainly never have happened without the help and encouragement of the folks at Cambridge University Press, particularly Lara Zoble, who kept us on track throughout the writing and publication process.