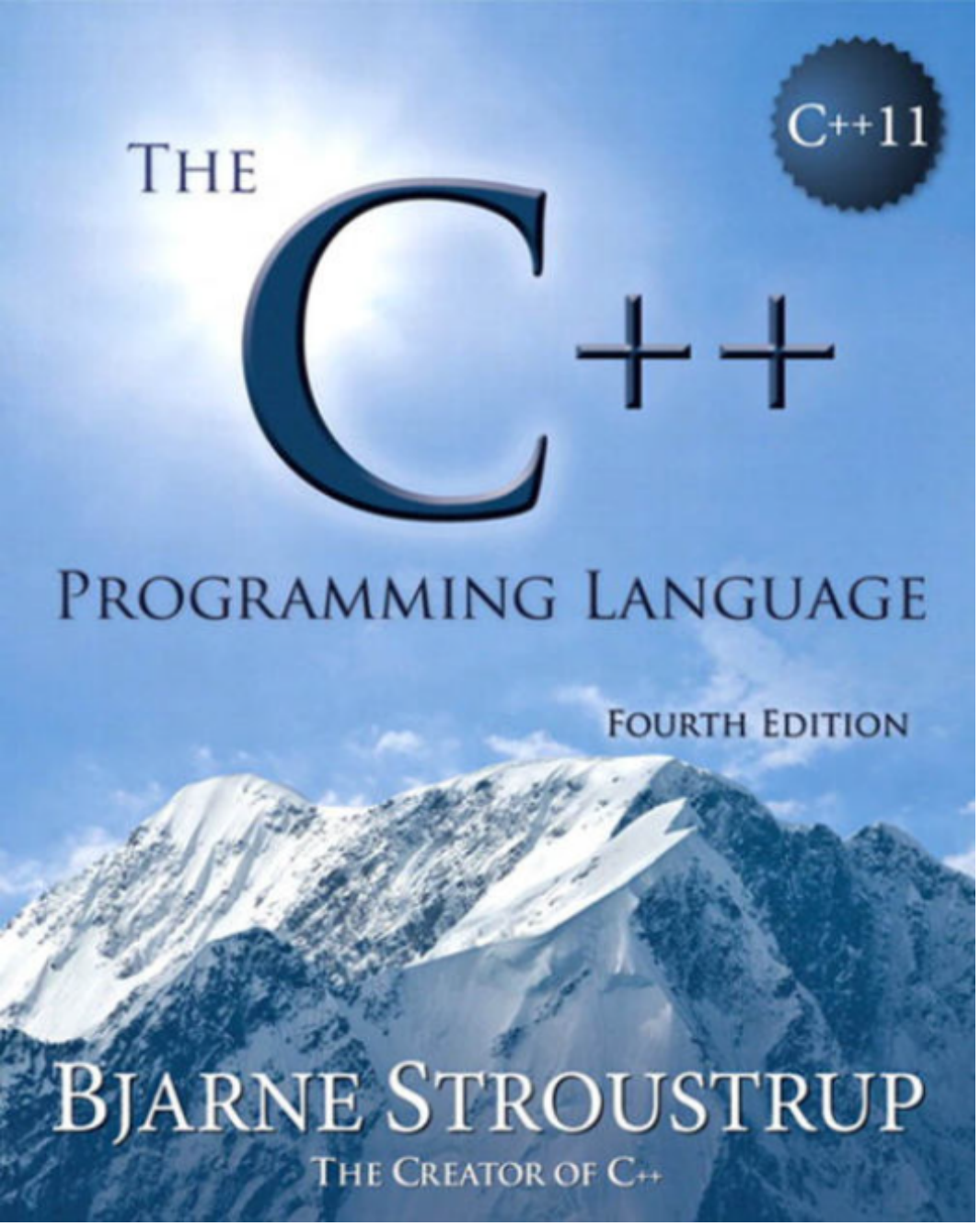
C++11
THE
C++
PROGRAMMING LANGUAGE
FOURTH EDITION
BJARNE STROUSTRUP
THE CREATOR OF C++

The C++ Programming Language

Fourth Edition

Bjarne Stroustrup

Library of Congress Cataloging-in-Publication Data

Stroustrup, Bjarne.
The C++ programming language / Bjarne Stroustrup.—Fourth edition.
pages cm
Includes bibliographical references and index.
ISBN 978-0-321-56384-2 (pbk. : alk. paper)—ISBN 0-321-56384-0 (pbk. : alk. paper)
1. C++ (Computer programming language) I. Title.

QA76.73.C153 S77 2013
005.13'3—dc23 2013002159

Co

Contents

Preface

Preface to the Fourth Edition v
Preface to the Third Edition ix
Preface to the Second Edition xi
Preface to the First Edition xii

Part I: Introductory Material

1. Notes to the Reader 3
2. A Tour of C++: The Basics 37
3. A Tour of C++: Abstraction Mechanisms 59
4. A Tour of C++: Containers and Algorithms 87

11. Select Operations 273
12. Functions 305
13. Exception Handling 343
14. Namespaces 389
15. Source Files and Programs 419

Part III: Abstraction Mechanisms

16. Classes 449
17. Construction, Cleanup, Copy, and Move 481
18. Overloading 527
19. Special Operators 549
20. Derived Classes 577
21. Class Hierarchies 613
22. Run-Time Type Information 641
23. Templates 665
24. Generic Programming 699
25. Specialization 721
26. Instantiation 741
27. Templates and Hierarchies 759
28. Metaprogramming 779
29. A Matrix Design 827

Part IV: The Standard Library

30. Standard Library Summary 859
31. STL Containers 885
32. STL Algorithms 927
33. STL Iterators 953
34. Memory and Resources 973
35. Utilities 1009
36. Strings 1033
37. Regular Expressions 1051
38. I/O Streams 1073

I

All problems in co
can be solved by another level
except for the problem of too many layers
– Da

C++ feels like a new language. That is, I can express my ideas more clearly, mo
more directly in C++11 than I could in C++98. Furthermore, the resulting progr
checked by the compiler and run faster.

In this book, I aim for *completeness*. I describe every language feature and st
component that a professional programmer is likely to need. For each, I provide:

- *Rationale*: What kinds of problems is it designed to help solve? What princ
  the design? What are the fundamental limitations?
- *Specification*: What is its definition? The level of detail is chosen for the ex
  mer; the aspiring language lawyer can follow the many references to the ISO s
- *Examples*: How can it be used well by itself and in combination with other fe
  are the key techniques and idioms? What are the implications for maintaina
  formance?

The use of C++ has changed dramatically over the years and so has the language its

techniques for writing quality code. C++ is a language for someone who takes the tas
ming seriously. Our civilization depends critically on software; it had better be qualit

There are billions of lines of C++ deployed. This puts a premium on stability
1995 C++ code still works and will continue to work for decades. However, for al
you can do better with modern C++; if you stick to older styles, you will be writing
and worse-performing code. The emphasis on stability also implies that standar
code you write today will still work a couple of decades from now. All code in this b
to the 2011 ISO C++ standard.

This book is aimed at three audiences:

- C++ programmers who want to know what the latest ISO C++ standard has to
- C programmers who wonder what C++ provides beyond C, and
- People with a background in application languages, such as Java, C#, Pyth
  looking for something "closer to the machine" – something more flexible, so
  ing better compile-time checking, or something offering better performance.

Naturally, these three groups are not disjoint – a professional software developer mas
just one programming language.

This book assumes that its readers are programmers. If you ask, "What's a
"What's a compiler?" then this book is not (yet) for you; instead, I recommend my
Principles and Practice Using C++ to get started with programming and C++.
assume that readers have some maturity as software developers. If you ask "Why bo
or say, "All languages are basically the same; just show me the syntax" or are confi
is a single language that is ideal for every task, this is not the book for you.

What features does C++11 offer over and above C++98? A machine model suital
computers with lots of concurrency. Language and standard-library facilities for d
level concurrent programming (e.g., using multicores). Regular expression hand
management pointers, random numbers, improved containers (including, hash tabl
General and uniform initialization, a simpler `for`-statement, move semantics, basic Un
lambdas, general constant expressions, control over class defaults, variadic template
literals, and more. Please remember that those libraries and language features exist t
gramming techniques for developing quality software. They are meant to be used in
as bricks in a building set – rather than to be used individually in relative isolation
cific problem. A computer is a universal machine, and C++ serves it in that capacity.
C++'s design aims to be sufficiently flexible and general to cope with future proble
of by its designers.

Acknowledgments

In addition to the people mentioned in the acknowledgment sections of the previ would like to thank Pete Becker, Hans-J. Boehm, Marshall Clow, Jonathan Coe, La Walter Daugherty, J. Daniel Garcia, Robert Harle, Greg Hickman, Howard H Kernighan, Daniel Krügler, Nevin Liber, Michel Michaud, Gary Powell, Jan Christia and Leor Zolman. Without their help this book would have been much poorer.

Thanks to Howard Hinnant for answering many questions about the standard libra

Andrew Sutton is the author of the Origin library, which was the testbed for much sion of emulating concepts in the template chapters, and of the matrix library that Chapter 29. The Origin library is open source and can be found by searching the Wel and "Andrew Sutton."

Thanks to my graduate design class for finding more problems with the "tour anyone else.

Had I been able to follow every piece of advice of my reviewers, the book woul have been much improved, but it would also have been hundreds of pages longer reviewer suggested adding technical details, advanced examples, and many usefu conventions; every novice reviewer (or educator) suggested adding examples; and observed (correctly) that the book may be too long.

Thanks to Princeton University's Computer Science Department, and especial Kernighan, for hosting me for part of the sabbatical that gave me time to write this bo

Thanks to Cambridge University's Computer Lab, and especially Prof. Andy Ho ing me for part of the sabbatical that gave me time to write this book.

Thanks to my editor, Peter Gordon, and his production team at Addison-Wesley and patience.

College Station, Texas *Bja*

This page intentionally left blank

Preface to the Third E

Programming is t
– K

I find using C++ more enjoyable than ever. C++'s support for design and prog
improved dramatically over the years, and lots of new helpful techniques have been
its use. However, C++ is not *just* fun. Ordinary practical programmers have achie
improvements in productivity, maintainability, flexibility, and quality in projects of
kind and scale. By now, C++ has fulfilled most of the hopes I originally had for it,
ceeded at tasks I hadn't even dreamt of.

This book introduces standard C++† and the key programming and design techni
by C++. Standard C++ is a far more powerful and polished language than the version
duced by the first edition of this book. New language features such as namespace
templates, and run-time type identification allow many techniques to be applied mor
was possible before, and the standard library allows the programmer to start from
level than the bare language.

About a third of the information in the second edition of this book came from
third edition is the result of a rewrite of even larger magnitude. It offers something to
experienced C++ programmer; at the same time, this book is easier for the novice to
its predecessors were. The explosion of C++ use and the massive amount of exper
lated as a result makes this possible.

The definition of an extensive standard library makes a difference to the way C++

be presented. As before, this book presents C++ independently of any particular in
and as before, the tutorial chapters present language constructs and concepts in a
order so that a construct is used only after it has been defined. However, it is much
well-designed library than it is to understand the details of its implementation. There
dard library can be used to provide realistic and interesting examples well before a
assumed to understand its inner workings. The standard library itself is also a fertile
gramming examples and design techniques.

This book presents every major C++ language feature and the standard library.
around language and library facilities. However, features are presented in the contex

† ISO/IEC 14882, Standard for the C++ Programming Language.