

教育部高等教育司推荐  
国外优秀信息科学与技术系列教学用书

# C++ 程序设计语言

(特别版 影印版)

## THE C++ PROGRAMMING LANGUAGE

(Special Edition)

■ Bjarne Stroustrup



高等教育出版社  
Higher Education Press

Pearson Education



出版集团

教育部高等教育司推荐  
国外优秀信息科学与技术系列教学用书

# C++程序设计语言

(特别版 影印版)

## THE C++ PROGRAMMING LANGUAGE

( Special Edition )

Bjarne Stroustrup



高等教育出版社



Pearson Education 出版集团

图字: 01-2001-2172 号

English Reprint Copyright © 2001 by PEARSON EDUCATION NORTH ASIA LIMIED and HIGHER  
EDUCATION PRESS

The C++ Programming Language: Special Edition, by Bjarne Stroustrup, Copyright © 2000

All Rights Reserved

Published by arrangement with the original publisher, Pearson Education Inc., publishing as ADDISON  
WESLEY LOGMAN, a Pearson Education company

This edition is authorized for sale only in the People's Republic of China (excluding the Special  
Administrative Regions of Hong Kong and Macau)

### 图书在版编目(CIP)数据

C++ 程序设计语言:特别版/(美)斯特劳施特鲁普  
(Stroustrup, B.).—影印本.—北京:高等教育  
出版社,2001.(2002 重印)

ISBN 7-04-010095-9

I. C… II. 斯… III. C 语言—程序设计—高等学  
校—教材—英文 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 045089 号

C++ 程序设计语言(特别版 影印版)

Bjarne Stroustrup

---

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010—64054588

传 真 010—64014048

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷 北京外文印刷厂

开 本 787×1092 1/16

版 次 2001 年 8 月影印版

印 张 64.5

印 次 2002 年 4 月第 3 次印刷

字 数 1 536 000

定 价 55.00 元

---

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

# 前 言

20 世纪末,以计算机和通信技术为代表的信息科学和技术,对世界的经济、军事、科技、教育、文化、卫生等方面的发展产生了深刻的影响,由此而兴起的信息产业已经成为世界经济发展的支柱。进入 21 世纪,各国为了加快本国的信息产业,加大了资金投入和政策扶持。

为了加快我国信息产业的进程,在我国《国民经济和社会发展第十个五年计划纲要》中,明确提出“以信息化带动工业化,发挥后发优势,实现社会生产力的跨越式发展。”信息产业的国际竞争将日趋激烈。在我国加入 WTO 后,我国信息产业将面临国外竞争对手的严峻挑战。竞争成败最终将取决于信息科学和技术人才的多少与优劣。

在 20 世纪末,我国信息产业虽然得到迅猛发展,但与国际先进国家相比,差距还很大。为了赶上并超过国际先进水平,我国必须加快信息技术人才的培养,特别要培养一大批具有国际竞争能力的高水平的信息技术人才,促进我国信息产业和国家信息化水平的全面提高。为此,教育部高等教育司根据教育部吕福源副部长的意见,在长期重视推动高等学校信息科学和技术教学的基础上,将实施超前发展战略,采取一些重要举措,加快推动高等学校的信息科学和技术等相关专业的教学工作。在大力宣传、推荐我国专家编著的面向 21 世纪和“九五”重点的信息科学和技术课程教材的基础上,在有条件的高等学校的某些信息科学和技术课程中推动使用国外优秀教材的影印版进行英语或双语教学,以缩短我国在计算机教学上与国际先进水平的差距,同时也有助于强化我国大学生的英语水平。

为了达到上述目的,在分析一些出版社已影印相关教材,一些学校已试用影印教材进行教学的基础上,教育部高等教育司组织并委托高等教育出版社开展国外优秀信息科学和技术优秀教材及其教学辅助材料的引进研究与影印出版的试点工作。为推动用影印版教材进行教学创造条件。

本次引进的系列教材的影印出版工作,是在对我国高校信息科学和技术专业的课程与美国高校的对比分析的基础上展开的;所影印出版的教材均由我国主要高校

的信息科学和技术专家组成的专家组，从国外近两年出版的大量最新教材中精心筛选评审通过的内容新、有影响的优秀教材；影印教材的定价原则上应与我国大学教材价格相当。

教育部高等教育司将此影印系列教材推荐给高等学校，希望有关教师选用，使用后有什么意见和建议请及时反馈。也希望有条件的出版社，根据影印教材的要求，积极参加此项工作，以便引进更多、更新、更好的外国教材和教学辅助材料。

同时，感谢国外有关出版公司对此项引进工作的配合，欢迎更多的国外公司关心并参与此项工作。

教育部高等教育司

二〇〇一年四月

# Preface

*Programming is understanding.*  
– Kristen Nygaard

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

This book introduces standard C++<sup>†</sup> and the key programming and design techniques supported by C++. Standard C++ is a far more powerful and polished language than the version of C++ introduced by the first edition of this book. New language features such as namespaces, exceptions, templates, and run-time type identification allow many techniques to be applied more directly than was possible before, and the standard library allows the programmer to start from a much higher level than the bare language.

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

The definition of an extensive standard library makes a difference to the way C++ concepts can be presented. As before, this book presents C++ independently of any particular implementation, and as before, the tutorial chapters present language constructs and concepts in a “bottom up” order so that a construct is used only after it has been defined. However, it is much easier to use a well-designed library than it is to understand the details of its implementation. Therefore, the standard library can be used to provide realistic and interesting examples well before a reader can be assumed to understand its inner workings. The standard library itself is also a fertile source of programming examples and design techniques.

---

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

This book presents every major C++ language feature and the standard library. It is organized around language and library facilities. However, features are presented in the context of their use. That is, the focus is on the language as the tool for design and programming rather than on the language in itself. This book demonstrates key techniques that make C++ effective and teaches the fundamental concepts necessary for mastery. Except where illustrating technicalities, examples are taken from the domain of systems software. A companion, *The Annotated C++ Language Standard*, presents the complete language definition together with annotations to make it more comprehensible.

The primary aim of this book is to help the reader understand how the facilities offered by C++ support key programming techniques. The aim is to take the reader far beyond the point where he or she gets code running primarily by copying examples and emulating programming styles from other languages. Only a good understanding of the ideas behind the language facilities leads to mastery. Supplemented by implementation documentation, the information provided is sufficient for completing significant real-world projects. The hope is that this book will help the reader gain new insights and become a better programmer and designer.

### Acknowledgments

In addition to the people mentioned in the acknowledgement sections of the first and second editions, I would like to thank Matt Austern, Hans Boehm, Don Caldwell, Lawrence Cowl, Alan Feuer, Andrew Forrest, David Gay, Tim Griffin, Peter Juhl, Brian Kernighan, Andrew Koenig, Mike Mowbray, Rob Murray, Lee Nackman, Joseph Newcomer, Alex Stepanov, David Vandevorde, Peter Weinberger, and Chris Van Wyk for commenting on draft chapters of this third edition. Without their help and suggestions, this book would have been harder to understand, contained more errors, been slightly less complete, and probably been a little bit shorter.

I would also like to thank the volunteers on the C++ standards committees who did an immense amount of constructive work to make C++ what it is today. It is slightly unfair to single out individuals, but it would be even more unfair not to mention anyone, so I'd like to especially mention Mike Ball, Dag Brück, Sean Corfield, Ted Goldstein, Kim Knuttila, Andrew Koenig, José Lajoie, Dmitry Lenkov, Nathan Myers, Martin O'Riordan, Tom Plum, Jonathan Shopiro, John Spicer, Jerry Schwarz, Alex Stepanov, and Mike Vilot, as people who each directly cooperated with me over some part of C++ and its standard library.

After the initial printing of this book, many dozens of people have mailed me corrections and suggestions for improvements. I have been able to accommodate many of their suggestions within the framework of the book so that later printings benefitted significantly. Translators of this book into many languages have also provided many clarifications. In response to requests from readers, I have added appendices D and E. Let me take this opportunity to thank a few of those who helped: Dave Abrahams, Matt Austern, Jan Bielawski, Janina Mincer Daszkiewicz, Andrew Koenig, Dietmar Kühl, Nicolai Josuttis, Nathan Myers, Paul E. Sevinç, Andy Tenne-Sens, Shoichi Uchida, Ping-Fai (Mike) Yang, and Dennis Yelle.

*Murray Hill, New Jersey*

*Bjarne Stroustrup*



# Preface to the Second Edition

*The road goes ever on and on.*

– *Bilbo Baggins*

As promised in the first edition of this book, C++ has been evolving to meet the needs of its users. This evolution has been guided by the experience of users of widely varying backgrounds working in a great range of application areas. The C++ user-community has grown a hundredfold during the six years since the first edition of this book; many lessons have been learned, and many techniques have been discovered and/or validated by experience. Some of these experiences are reflected here.

The primary aim of the language extensions made in the last six years has been to enhance C++ as a language for data abstraction and object-oriented programming in general and to enhance it as a tool for writing high-quality libraries of user-defined types in particular. A “high-quality library,” is a library that provides a concept to a user in the form of one or more classes that are convenient, safe, and efficient to use. In this context, *safe* means that a class provides a specific type-safe interface between the users of the library and its providers; *efficient* means that use of the class does not impose significant overheads in run-time or space on the user compared with hand-written C code.

This book presents the complete C++ language. Chapters 1 through 10 give a tutorial introduction; Chapters 11 through 13 provide a discussion of design and software development issues; and, finally, the complete C++ reference manual is included. Naturally, the features added and resolutions made since the original edition are integral parts of the presentation. They include refined overloading resolution, memory management facilities, and access control mechanisms, type-safe linkage, *const* and *static* member functions, abstract classes, multiple inheritance, templates, and exception handling.

C++ is a general-purpose programming language; its core application domain is systems programming in the broadest sense. In addition, C++ is successfully used in many application areas that are not covered by this label. Implementations of C++ exist from some of the most modest microcomputers to the largest supercomputers and for almost all operating systems. Consequently, this book describes the C++ language itself without trying to explain a particular implementation, programming environment, or library.

This book presents many examples of classes that, though useful, should be classified as “toys.” This style of exposition allows general principles and useful techniques to stand out more



clearly than they would in a fully elaborated program, where they would be buried in details. Most of the useful classes presented here, such as linked lists, arrays, character strings, matrices, graphics classes, associative arrays, etc., are available in “bulletproof” and/or “goldplated” versions from a wide variety of commercial and non-commercial sources. Many of these “industrial strength” classes and libraries are actually direct and indirect descendants of the toy versions found here.

This edition provides a greater emphasis on tutorial aspects than did the first edition of this book. However, the presentation is still aimed squarely at experienced programmers and endeavors not to insult their intelligence or experience. The discussion of design issues has been greatly expanded to reflect the demand for information beyond the description of language features and their immediate use. Technical detail and precision have also been increased. The reference manual, in particular, represents many years of work in this direction. The intent has been to provide a book with a depth sufficient to make more than one reading rewarding to most programmers. In other words, this book presents the C++ language, its fundamental principles, and the key techniques needed to apply it. Enjoy!

### **Acknowledgments**

In addition to the people mentioned in the acknowledgements section in the preface to the first edition, I would like to thank Al Aho, Steve Buroff, Jim Coplien, Ted Goldstein, Tony Hansen, Lorraine Juhl, Peter Juhl, Brian Kernighan, Andrew Koenig, Bill Leggett, Warren Montgomery, Mike Mowbray, Rob Murray, Jonathan Shopiro, Mike Vilot, and Peter Weinberger for commenting on draft chapters of this second edition. Many people influenced the development of C++ from 1985 to 1991. I can mention only a few: Andrew Koenig, Brian Kernighan, Doug McIlroy, and Jonathan Shopiro. Also thanks to the many participants of the “external reviews” of the reference manual drafts and to the people who suffered through the first year of X3J16.

*Murray Hill, New Jersey*

*Bjarne Stroustrup*

# Preface to the First Edition

*Language shapes the way we think,  
and determines what we can think about.*

– B.L. Whorf

C++ is a general purpose programming language designed to make programming more enjoyable for the serious programmer. Except for minor details, C++ is a superset of the C programming language. In addition to the facilities provided by C, C++ provides flexible and efficient facilities for defining new types. A programmer can partition an application into manageable pieces by defining new types that closely match the concepts of the application. This technique for program construction is often called *data abstraction*. Objects of some user-defined types contain type information. Such objects can be used conveniently and safely in contexts in which their type cannot be determined at compile time. Programs using objects of such types are often called *object based*. When used well, these techniques result in shorter, easier to understand, and easier to maintain programs.

The key concept in C++ is *class*. A class is a user-defined type. Classes provide data hiding, guaranteed initialization of data, implicit type conversion for user-defined types, dynamic typing, user-controlled memory management, and mechanisms for overloading operators. C++ provides much better facilities for type checking and for expressing modularity than C does. It also contains improvements that are not directly related to classes, including symbolic constants, inline substitution of functions, default function arguments, overloaded function names, free store management operators, and a reference type. C++ retains C's ability to deal efficiently with the fundamental objects of the hardware (bits, bytes, words, addresses, etc.). This allows the user-defined types to be implemented with a pleasing degree of efficiency.

C++ and its standard libraries are designed for portability. The current implementation will run on most systems that support C. C libraries can be used from a C++ program, and most tools that support programming in C can be used with C++.

This book is primarily intended to help serious programmers learn the language and use it for nontrivial projects. It provides a complete description of C++, many complete examples, and many more program fragments.

### **Acknowledgments**

C++ could never have matured without the constant use, suggestions, and constructive criticism of many friends and colleagues. In particular, Tom Cargill, Jim Coplien, Stu Feldman, Sandy Fraser, Steve Johnson, Brian Kernighan, Bart Locanthi, Doug McIlroy, Dennis Ritchie, Larry Rosler, Jerry Schwarz, and Jon Shopiro provided important ideas for development of the language. Dave Presotto wrote the current implementation of the stream I/O library.

In addition, hundreds of people contributed to the development of C++ and its compiler by sending me suggestions for improvements, descriptions of problems they had encountered, and compiler errors. I can mention only a few: Gary Bishop, Andrew Hume, Tom Karzes, Victor Milenkovic, Rob Murray, Leonie Rose, Brian Schmult, and Gary Walker.

Many people have also helped with the production of this book, in particular, Jon Bentley, Laura Eaves, Brian Kernighan, Ted Kowalski, Steve Mahaney, Jon Shopiro, and the participants in the C++ course held at Bell Labs, Columbus, Ohio, June 26-27, 1985.

*Murray Hill, New Jersey*

*Bjarne Stroustrup*

责任编辑 张海波  
封面设计 韩 冲  
责任印制 陈伟光

# Contents

<b>Preface</b>	<b>v</b>
<b>Preface to Second Edition</b>	<b>vii</b>
<b>Preface to First Edition</b>	<b>ix</b>
<b>Introductory Material</b>	<b>1</b>
1 Notes to the Reader .....	3
2 A Tour of C++ .....	21
3 A Tour of the Standard Library .....	45
<b>Part I: Basic Facilities</b>	<b>67</b>
4 Types and Declarations .....	69
5 Pointers, Arrays, and Structures .....	87
6 Expressions and Statements .....	107
7 Functions .....	143
8 Namespaces and Exceptions .....	165
9 Source Files and Programs .....	197

<b>Part II: Abstraction Mechanisms</b>	<b>221</b>
10 Classes .....	223
11 Operator Overloading .....	261
12 Derived Classes .....	301
13 Templates .....	327
14 Exception Handling .....	355
15 Class Hierarchies .....	389
 <b>Part III: The Standard Library</b>	 <b>427</b>
16 Library Organization and Containers .....	429
17 Standard Containers .....	461
18 Algorithms and Function Objects .....	507
19 Iterators and Allocators .....	549
20 Strings .....	579
21 Streams .....	605
22 Numerics .....	657
 <b>Part IV: Design Using C++</b>	 <b>689</b>
23 Development and Design .....	691
24 Design and Programming .....	723
25 Roles of Classes .....	765
 <b>Appendices</b>	 <b>791</b>
A The C++ Grammar .....	793
B Compatibility .....	815
C Technicalities .....	827
D Locales .....	869
E Standard Library Exception Safety .....	935
 <b>Index</b>	 <b>969</b>

# Introduction

This introduction gives an overview of the major concepts and features of the C++ programming language and its standard library. It also provides an overview of this book and explains the approach taken to the description of the language facilities and their use. In addition, the introductory chapters present some background information about C++, the design of C++, and the use of C++.

## Chapters

- 1 Notes to the Reader
- 2 A Tour of C++
- 3 A Tour of the Standard Library



“... and you, Marcus, you have given me many things; now I shall give you this good advice. Be many people. Give up the game of being always Marcus Coccoza. You have worried too much about Marcus Coccoza, so that you have been really his slave and prisoner. You have not done anything without first considering how it would affect Marcus Coccoza’s happiness and prestige. You were always much afraid that Marcus might do a stupid thing, or be bored. What would it really have mattered? All over the world people are doing stupid things ... I should like you to be easy, your little heart to be light again. You must from now, be more than one, many people, as many as you can think of ...”

– Karen Blixen

(“The Dreamers” from “Seven Gothic Tales”

written under the pseudonym Isak Dinesen,

Random House, Inc.

Copyright, Isak Dinesen, 1934 renewed 1961)

---

## Notes to the Reader

---

*"The time has come," the Walrus said,  
"to talk of many things."  
— L. Carroll*

Structure of this book — how to learn C++ — the design of C++ — efficiency and structure — philosophical note — historical note — what C++ is used for — C and C++ — suggestions for C programmers — suggestions for C++ programmers — thoughts about programming in C++ — advice — references.

### 1.1 The Structure of This Book

This book consists of six parts:

Introduction: Chapters 1 through 3 give an overview of the C++ language, the key programming styles it supports, and the C++ standard library.

Part I: Chapters 4 through 9 provide a tutorial introduction to C++'s built-in types and the basic facilities for constructing programs out of them.

Part II: Chapters 10 through 15 are a tutorial introduction to object-oriented and generic programming using C++.

Part III: Chapters 16 through 22 present the C++ standard library.

Part IV: Chapters 23 through 25 discuss design and software development issues.

Appendices: Appendices A through E provide language-technical details.

Chapter 1 provides an overview of this book, some hints about how to use it, and some background information about C++ and its use. You are encouraged to skim through it, read what appears interesting, and return to it after reading other parts of the book.

Chapters 2 and 3 provide an overview of the major concepts and features of the C++ programming language and its standard library. Their purpose is to motivate you to spend time on fundamental concepts and basic language features by showing what can be expressed using the complete