

Mc
Graw
Hill Education

大学计算机教育国外著名教材系列 (影印版)

C++:THE COMPLETE REFERENCE

FOURTH EDITION

C++完全参考 手册 (第4版)



Herbert Schildt 著



清华大学出版社

大学计算机教育国外著名教材系列(影印版)

C++ : The Complete Reference

Fourth Edition

C++ 完全参考手册

第4版

Herbert Schildt

江苏工业学院图书馆
藏书章

清华大学出版社

北京

Herbert Schildt

C++ : The Complete Reference, Fourth Edition

EISBN: 0-07-222680-3

Copyright © 2003 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Authorized English language edition jointly published by McGraw-Hill Education (Asia) Co. and Tsinghua University Press. This edition is authorized for sale only to the educational and training institutions, and within the territory of the People's Republic of China (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书英文影印版由清华大学出版社和美国麦格劳-希尔教育出版(亚洲)公司合作出版。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾地区)针对教育及培训机构之销售。未经许可之出口,视为违反著作权法,将受法律之制裁。

未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字:01-2004-5638

版权所有,翻印必究。举报电话:010-62782989 13801310933

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C++ 完全参考手册 = C++ : The Complete Reference, 第4版/希尔德特(Schildt, H.)著. —影印本.

—北京:清华大学出版社,2005.1

(大学计算机教育国外著名教材系列)

ISBN 7-302-10157-4

I. C… II. 希… 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(2004)第135499号

出版者:清华大学出版社

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

社总机:010-62770175

地址:北京清华大学学研大厦

邮编:100084

客户服务:010-62776969

责任编辑:李晔

印刷者:清华大学印刷厂

装订者:三河市金元装订厂

发行者:新华书店总店北京发行所

开本:185×230 印张:65.25

版次:2005年1月第1版 2005年1月第1次印刷

书号:ISBN 7-302-10157-4/TP·1069

印数:1~3000

定价:96.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

出版说明

进入 21 世纪, 世界各国的经济、科技以及综合国力的竞争将更加激烈。竞争的中心无疑是对人才的竞争。谁拥有大量高素质的人才, 谁就能在竞争中取得优势。高等教育, 作为培养高素质人才的事业, 必然受到高度重视。目前我国高等教育的教材更新较慢, 为了加快教材的更新频率, 教育部正在大力促进我国高校采用国外原版教材。

清华大学出版社从 1996 年开始, 与国外著名出版公司合作, 影印出版了“大学计算机教育丛书(影印版)”等一系列引进图书, 受到国内读者的欢迎和支持。跨入 21 世纪, 我们本着为我国高等教育教材建设服务的初衷, 在已有的基础上, 进一步扩大选题内容, 改变图书开本尺寸, 一如既往地请有关专家挑选适用于我国高等本科及研究生计算机教育的国外经典教材或著名教材, 组成本套“大学计算机教育国外著名教材系列(影印版)”, 以飨读者。深切期盼读者及时将使用本系列教材的效果和意见反馈给我们。更希望国内专家、教授积极向我们推荐国外计算机教育的优秀教材, 以利我们把“大学计算机教育国外著名教材系列(影印版)”做得更好, 更适合高校师生的需要。

清华大学出版社

Introduction

If there is one language that defines modern programming, it is C++. Its syntax, style, and philosophy have set the standard by which all other languages are judged. Furthermore, C++ is the universal language of programming. When an algorithm or technique is described, it is usually done so using the C++ syntax. The long-term success of C++ has also left a lasting impression on computer language development. For example, both Java and C# are descended from C++. Frankly, to be a professional programmer implies proficiency in C++. It is the one language that no programmer can afford to ignore.

This is the fourth edition of C++: *The Complete Reference*. It fully describes and demonstrates the keywords, syntax, functions, classes, and features that define the C++ language. More specifically, this book fully describes Standard C++. This is the version of C++ defined by the ANSI/ISO Standard for C++ and it is the version of C++ that is supported by all major compilers, including Microsoft's Visual C++ and Borland's C++ Builder. Thus, the information in this book is applicable to all modern programming environments.

In the time that has passed since the previous edition of this book, there have been no changes to the C++ language. There have, however, been big changes to the computing environment. For example, a new standard for C, called C99, was created, Java became the dominant language for Web programming, the .NET Framework was

released, and C# was invented. Through all the changes of the past few years, one thing has remained constant: the staying power of C++. C++ has been, is, and will remain the preeminent language for the development of high-performance software well into the foreseeable future.

What's New in the Fourth Edition

The overall structure and organization of the fourth edition is similar to the third edition. Thus, if you have been using the third edition, you will feel right at home with the fourth edition. Most of the changes to the fourth edition involve updating and expanding the coverage throughout. In some cases, additional details were added. In other cases, the presentation of a topic was improved. In still other situations, descriptions were modernized to reflect the current programming environment. Several new sections were also added. In Part One, the relationship of C++ to the new C standard, called C99, is noted where appropriate.

Two appendices were also added. The first described the extended keywords defined by Microsoft that are used for creating managed code for the .NET Framework. The second shows off an area of personal interest: robotics. Robotics has long been a hobby of mine and I thought that many readers would find my experimental robot to be of interest. Most of the software that drives it is, of course, written in C++!

Finally, all code examples were retested against the current crop of compilers, including Microsoft's Visual Studio .NET and Borland's C++ Builder.

What's Inside

This book covers in detail all aspects of the C++ language, including its foundation, C. The book is divided into these five parts:

- The C Subset—The foundation of C++
- The C++ language
- The Standard Function Library
- The Standard Class Library
- Sample C++ applications

Part One provides a comprehensive discussion of the C subset of C++. As most readers will know, C is the foundation upon which C++ was built. It is the C subset that defines the bedrock features of C++, including such things as **for** loops and **if** statements. It also defines the essential nature of C++'s block structure, pointers, and functions. Since many readers are already familiar with and proficient in C, discussing the C subset separately in Part One prevents the knowledgeable C programmer from having to "wade through" reams of information he or she already knows. Instead, the

experienced C programmer can simply turn to the sections of this book that cover the C++-specific features.

Part Two discusses in detail the features that move beyond the C foundation and define the C++ language. These include its object-oriented features such as classes, constructors, destructors, RTTI, and templates. Thus, Part Two covers those constructs that “make C++, C++.”

Part Three describes the standard function library and Part Four examines the standard class library, including the STL (Standard Template Library). Part Five shows two practical examples of applying C++ and object-oriented programming.



A Book for All Programmers

This C++ reference is designed for all C++ programmers, regardless of their experience level. It does assume, however, a reader able to create at least a simple program. If you are just learning C++, this book will make an excellent companion to any C++ tutorial and serve as a source of answers to your specific questions. Experienced C++ pros will find the in-depth coverage of C++’s more advanced features especially useful.



If You’re Using Windows

If your computer uses Windows, then you have chosen the right language. C++ is completely at home with Windows programming. However, none of the programs in this book are Windows programs. Instead, they are console-based programs. The reason for this is easy to understand: Windows programs are, by their nature, large and complex. The overhead required to create even a minimal Windows skeletal program is 50 to 70 lines of code. To write Windows programs that demonstrate the features of C++ would require hundreds of lines of code each. Put simply, Windows is not an appropriate environment in which to discuss the features of a programming language. However, you can still use a Windows-based compiler to compile the programs in this book because the compiler will automatically create a console session in which to execute your program.



Don’t Forget: Code on the Web

Remember, the source code for all of the programs in this book is available free-of-charge on the Web at www.osborne.com. Downloading this code prevents you from having to type in the examples.

For Further Study

C++: *The Complete Reference* is your gateway to the Herb Schildt series of programming books. Here are some others that you will find of interest.

To learn more about C++, try

C++: A Beginner's Guide

C++ from the Ground Up

Teach Yourself C++

STL Programming from the Ground Up

C++ Programmer's Reference

To learn about Java programming, we recommend the following:

Java 2: A Beginner's Guide

Java 2: The Complete Reference

Java 2 Programmer's Reference

To learn about C#, Herb offers these books:

C#: A Beginner's Guide

C#: The Complete Reference

To learn about Windows programming we suggest the following Schildt books:

Windows 98 Programming from the Ground Up

Windows 2000 Programming from the Ground Up

MFC Programming from the Ground Up

The Windows Programming Annotated Archives

If you want to learn about the C language, which is the foundation of all modern programming, then the following titles will be of interest.

C: The Complete Reference

Teach Yourself C

**When you need solid answers, fast, turn to Herbert Schildt,
the recognized authority on programming.**

Contents

Introduction	xxiii
--------------------	-------

Part I

The Foundation of C++: The C Subset

	1	An Overview of C	3
		The Origins and History of C	4
		C Is a Middle-Level Language	5
		C Is a Structured Language	6
		C Is a Programmer's Language	8
		The Form of a C Program	9
		The Library and Linking	10
		Separate Compilation	12
		Understanding the .C and .CPP File Extensions	12
	2	Expressions	13
		The Five Basic Data Types	14
		Modifying the Basic Types	15
		Identifier Names	16
		Variables	17
		Where Variables Are Declared	18
		Local Variables	18

Formal Parameters	21
Global Variables	21
The const and volatile Qualifiers	23
const	23
volatile	24
Storage Class Specifiers	25
extern	25
static Variables	28
register Variables	30
Variable Initializations	31
Constants	32
Hexadecimal and Octal Constants	33
String Constants	33
Backslash Character Constants	33
Operators	35
The Assignment Operator	35
Type Conversion in Assignments	35
Multiple Assignments	37
Arithmetic Operators	37
Increment and Decrement	38
Relational and Logical Operators	40
Bitwise Operators	42
The ? Operator	47
The & and * Pointer Operators	48
The Compile-Time Operator sizeof	50
The Comma Operator	50
The Dot (.) and Arrow (->) Operators	51
The [] and () Operators	51
Precedence Summary	52
Expressions	52
Order of Evaluation	52
Type Conversion in Expressions	53
Casts	54
Spacing and Parentheses	55
Compound Assignments	56
3 Statements	57
True and False in C and C++	58
Selection Statements	59
if	59
Nested ifs	60
The if-else-if Ladder	62
The ? Alternative	63
The Conditional Expression	66



switch	67
Nested switch Statements	70
Iteration Statements	70
The for Loop	70
for Loop Variations	72
The Infinite Loop	76
for Loops with No Bodies	77
The while Loop	77
The do-while Loop	79
Declaring Variables Within Selection and Iteration Statements ...	81
Jump Statements	82
The return Statement	82
The goto Statement	83
The break Statement	83
The exit() Function	85
The continue Statement	86
Expression Statements	88
Block Statements	88
4 Arrays and Null-Terminated Strings	89
Single-Dimension Arrays	90
Generating a Pointer to an Array	92
Passing Single-Dimension Arrays to Functions	92
Null-Terminated Strings	94
Two-Dimensional Arrays	96
Arrays of Strings	100
Multidimensional Arrays	101
Indexing Pointers	102
Array Initialization	105
Unsize Array Initializations	106
A Tic-Tac-Toe Example	108
5 Pointers	113
What Are Pointers?	114
Pointer Variables	115
The Pointer Operators	115
Pointer Expressions	116
Pointer Assignments	117
Pointer Arithmetic	117
Pointer Comparisons	119
Pointers and Arrays	121
Arrays of Pointers	122
Multiple Indirection	123
Initializing Pointers	125
Pointers to Functions	126

C's Dynamic Allocation Functions	129
Problems with Pointers	131
6 Functions	137
The General Form of a Function	138
Scope Rules of Functions	138
Function Arguments	139
Call by Value, Call by Reference	139
Creating a Call by Reference	140
Calling Functions with Arrays	142
argc and argv—Arguments to main()	144
The return Statement	147
Returning from a Function	148
Returning Values	149
Returning Pointers	151
Functions of Type void	152
What Does main() Return?	153
Recursion	153
Function Prototypes	155
Standard Library Function Prototypes	157
Declaring Variable-Length Parameter Lists	158
Old-Style Versus Modern FunctionParameter Declarations	158
7 Structures, Unions, Enumerations, and User-Defined Types	161
Structures	162
Accessing Structure Members	165
Structure Assignments	165
Arrays of Structures	166
Passing Structures to Functions	167
Passing Structure Members to Functions	167
Passing Entire Structures to Functions	168
Structure Pointers	169
Declaring a Structure Pointer	170
Using Structure Pointers	170
Arrays and Structures Within Structures	173
Bit-Fields	174
Unions	176
Enumerations	180
Using sizeof to Ensure Portability	183
typedef	184
8 C-Style Console I/O	187
An Important Application Note	188

Reading and Writing Characters	189
A Problem with <code>getchar()</code>	190
Alternatives to <code>getchar()</code>	190
Reading and Writing Strings	192
Formatted Console I/O	195
<code>printf()</code>	195
Printing Characters	196
Printing Numbers	196
Displaying an Address	198
The <code>%n</code> Specifier	198
Format Modifiers	199
The Minimum Field Width Specifier	199
The Precision Specifier	200
Justifying Output	201
Handling Other Data Types	202
The <code>*</code> and <code>#</code> Modifiers	202
<code>scanf()</code>	203
Format Specifiers	203
Inputting Numbers	203
Inputting Unsigned Integers	205
Reading Individual Characters Using <code>scanf()</code>	205
Reading Strings	205
Inputting an Address	206
The <code>%n</code> Specifier	206
Using a Scanset	206
Discarding Unwanted White Space	207
Non-White-Space Characters in the Control String	208
You Must Pass <code>scanf()</code> Addresses	208
Format Modifiers	208
Suppressing Input	209
9 File I/O	211
C Versus C++ File I/O	212
Streams and Files	212
Streams	212
Text Streams	213
Binary Streams	213
Files	213
File System Basics	214
The File Pointer	215
Opening a File	215
Closing a File	217
Writing a Character	217
Reading a Character	218

Using fopen(), getc(), putc(), and fclose()	218
Using feof()	220
Working with Strings: fputs() and fgets()	222
rewind()	223
ferror()	224
Erasing Files	226
Flushing a Stream	226
fread() and fwrite()	227
Using fread() and fwrite()	227
fseek() and Random-Access I/O	229
fprintf() and fscanf()	230
The Standard Streams	232
The Console I/O Connection	233
Using freopen() to Redirect the Standard Streams	234
10 The Preprocessor and Comments	237
The Preprocessor	238
#define	238
Defining Function-like Macros	240
#error	241
#include	242
Conditional Compilation Directives	242
#if, #else, #elif, and #endif	243
#ifdef and #ifndef	245
#undef	246
Using defined	247
#line	248
#pragma	248
The # and ## Preprocessor Operators	248
Predefined Macro Names	250
Comments	250
Single-Line Comments	252

Part II

C++

11 An Overview of C++	255
The Origins of C++	256
What Is Object-Oriented Programming?	257
Encapsulation	258
Polymorphism	258
Inheritance	259
Some C++ Fundamentals	259
A Sample C++ Program	260

	A Closer Look at the I/O Operators	263
	Declaring Local Variables	264
	No Default to int	265
	The bool Data Type	266
	Old-Style vs. Modern C++	267
	The New C++ Headers	268
	Namespaces	269
	Working with an Old Compiler	270
	Introducing C++ Classes	270
	Function Overloading	275
	Operator Overloading	278
	Inheritance	278
	Constructors and Destructors	283
	The C++ Keywords	287
	The General Form of a C++ Program	288
	12 Classes and Objects	289
	Classes	290
	Structures and Classes Are Related	293
	Unions and Classes Are Related	295
	Anonymous Unions	296
	Friend Functions	297
	Friend Classes	302
	Inline Functions	303
	Defining Inline Functions Within a Class	306
	Parameterized Constructors	307
	Constructors with One Parameter: A Special Case	309
	Static Class Members	310
	Static Data Members	310
	Static Member Functions	315
	When Constructors and Destructors Are Executed	317
	The Scope Resolution Operator	319
	Nested Classes	319
	Local Classes	320
	Passing Objects to Functions	320
	Returning Objects	323
	Object Assignment	324
	13 Arrays, Pointers, References, and the Dynamic	
	Allocation Operators	325
	Arrays of Objects	326
	Creating Initialized vs. Uninitialized Arrays	328
	Pointers to Objects	329
	Type Checking C++ Pointers	332

The this Pointer	332
Pointers to Derived Types	334
Pointers to Class Members	337
References	339
Reference Parameters	339
Passing References to Objects	343
Returning References	344
Independent References	345
References to Derived Types	346
Restrictions to References	347
A Matter of Style	347
C++'s Dynamic Allocation Operators	347
Initializing Allocated Memory	349
Allocating Arrays	350
Allocating Objects	351
The nothrow Alternative	356
The Placement Form of new	357

14 Function Overloading, Copy Constructors, and Default Arguments	359
Function Overloading	360
Overloading Constructors	362
Overloading a Constructor to Gain Flexibility	362
Allowing Both Initialized and Uninitialized Objects	364
Copy Constructors	366
Finding the Address of an Overloaded Function	370
The overload Anachronism	371
Default Function Arguments	371
Default Arguments vs. Overloading	376
Using Default Arguments Correctly	377
Function Overloading and Ambiguity	378

15 Operator Overloading	383
Creating a Member Operator Function	384
Creating Prefix and Postfix Forms of the Increment and Decrement Operators	389
Overloading the Shorthand Operators	390
Operator Overloading Restrictions	390
Operator Overloading Using a Friend Function	391
Using a Friend to Overload ++ or --	393
Friend Operator Functions Add Flexibility	396
Overloading new and delete	398
Overloading new and delete for Arrays	403
Overloading the nothrow Version of new and delete	406

Overloading Some Special Operators	407
Overloading []	407
Overloading ()	411
Overloading ->	413
Overloading the Comma Operator	414
16 Inheritance	417
Base-Class Access Control	418
Inheritance and protected Members	420
Protected Base-Class Inheritance	424
Inheriting Multiple Base Classes	425
Constructors, Destructors, and Inheritance	426
When Constructors and Destructors Are Executed	426
Passing Parameters to Base-Class Constructors	430
Granting Access	434
Virtual Base Classes	437
17 Virtual Functions and Polymorphism	443
Virtual Functions	444
Calling a Virtual Function Through a Base	
Class Reference	447
The Virtual Attribute Is Inherited	448
Virtual Functions Are Hierarchical	450
Pure Virtual Functions	453
Abstract Classes	455
Using Virtual Functions	455
Early vs. Late Binding	458
18 Templates	459
Generic Functions	460
A Function with Two Generic Types	463
Explicitly Overloading a Generic Function	463
Overloading a Function Template	466
Using Standard Parameters with Template Functions	466
Generic Function Restrictions	467
Applying Generic Functions	468
A Generic Sort	469
Compacting an Array	470
Generic Classes	472
An Example with Two Generic Data Types	476
Applying Template Classes: A Generic Array Class	477
Using Non-Type Arguments with Generic Classes	479
Using Default Arguments with Template Classes	481
Explicit Class Specializations	483