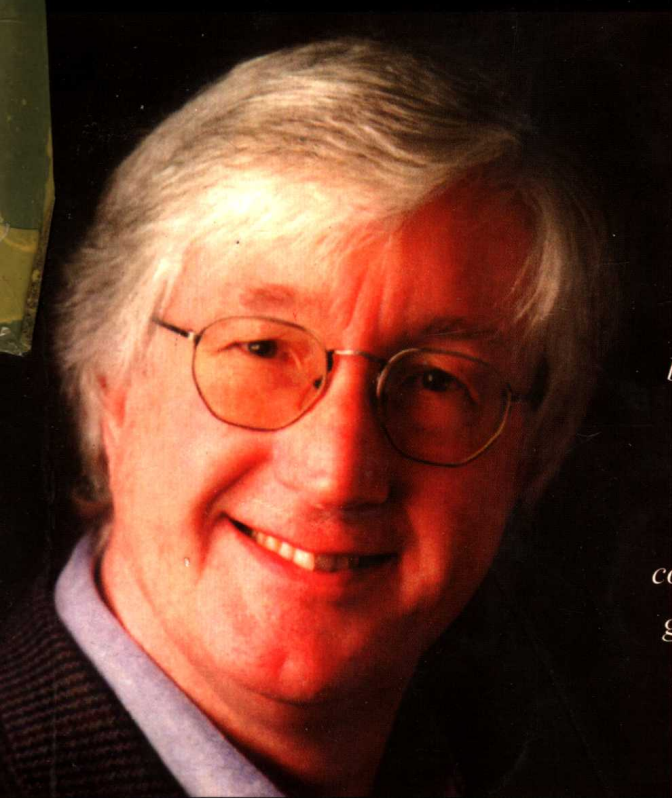


PROGRAMMER TO PROGRAMMER™

ANSI/ISO Compliant



"C++ is a big, powerful programming language; that's why it's the first choice of so many professionals. My book will provide you with a comprehensive, graduated tutorial. Each chapter builds on what you have learned, and every key feature is illustrated with complete working examples.

With this book, an up-to-date C++ compiler, and commitment and enthusiasm, you have all you need to get a solid understanding of C++. I'll show you how to gain the confidence and knowledge to apply it to your real-world problems."

Beginning C++ The Complete Language

C++ 入门经典

(美) Ivor Horton 著
李予敏 译



清华大学出版社

C++入门经典

(美) Ivor Horton 著

李 予 敏 译

清华大学出版社

北 京

内 容 简 介

C++在几乎所有的计算环境中都非常普及,而且可以用于几乎所有的应用程序。C++从C中继承了过程编程方式的高效性,并集成了面向对象编程的功能。C++在其标准库中提供了大量的功能。有许多商业C++库支持数量众多的操作系统环境 and 专业应用程序。但因为它的内容太多了,所以掌握C++并不十分容易。本书详尽论述了C++语言的各个方面,包括数据类型、程序控制、函数、指针、调试、类、重载、继承、多态性、模板、异常和输入输出等内容。每一章都以前述内容为基础,每个关键点都用具体的示例进行详细的讲解。

本书基本不需要读者具备任何C++知识,书中包含了理解C++的所有必要知识,读者可以从头开始编写自己的C++程序。本书也适合于具备另一种语言编程经验但希望全面掌握C++语言的读者。

EISBN: 1-861000-12-X

Beginning C++: The Complete Language

Ivor Horton

Copyright©1998 by Wrox Press Ltd.

Original English language Edition Published by Wrox Press Ltd.

All Rights Reserved.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号:图字:01-2002-3189

图书在版编目(CIP)数据

C++入门经典/(美)霍登(Horton, I.)著;李予敏译.—北京:清华大学出版社,2004

书名原文:Beginning C++: The Complete Language

ISBN 7-302-08402-5

I. C… II. ①霍…②李… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2004)第026831号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:曹 康

文稿编辑:于 平

封面设计:康 博

版式设计:康 博

印 刷 者:清华大学印刷厂

装 订 者:三河市化甲屯小学装订二厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:52.75 字数:1350千字

版 次:2004年5月第1版 2004年5月第1次印刷

书 号:ISBN 7-302-08402-5/TP·6041

印 数:1~4000

定 价:108.00元

出版者的话

随着国际互联网的快速崛起和迅猛发展，计算机之间的互联需求越来越迫切，而目前计算机硬件设备的不兼容性严重束缚了互联网的发展，引发了新一轮的跨平台软件的开发浪潮。软件商纷纷推出新的战略规划和解决方案，Microsoft 提出的.NET 战略就是其中的经典之作。

在经历这场浪潮的洗涤和考验过程中，全球的软件开发人员都迫切需要了解新的软件技术和开发思路。为了满足国内 IT 从业人员的需要，清华大学出版社从 Wrox 出版公司引进了若干套编程系列丛书，“入门经典”系列是其中不可或缺的入门之作。作为世界著名的编程技术图书的出版公司，Wrox 推出的这套“入门经典”系列丛书主要面向编程的初学者、需要了解.NET 策略的程序员，以及需要迅速掌握多门编程工具的程序员。该丛书依旧秉承了 Wrox 公司“由程序员为程序员而著（Programmer to Programmer）”的创作理念，每本书均由世界顶级的编程高手执笔。他们站在资深程序员的高度，循序渐进地为初学者讲述了.NET 的理念和构架、编程基本思想、编程语言基础、程序的控制和调试、Windows 应用程序的开发、对象编程技术、数据库访问技术、Web 程序开发和.NET 构架应用等最新的软件开发知识，同时辅以大量操作性强的程序为示例，为读者提供了清晰的编程思路和宝贵的编程经验。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码和获取相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前 言

本书主要介绍标准的 C++ 编程语言，涉及 C++ 的语法、入门、面向对象的功能和标准库等所有基本内容。阅读本书将获得编写 C++ 应用程序的所有必要知识。

为什么要学习 C++

C++ 自问世以来，已成为使用最广泛的一种编程语言，其用户量一直不断增加。C++ 由于其极高的灵活性和强大的功能，常常用于专业应用程序的开发，它要比许多人想像的更容易理解。只要有正确的引导，掌握 C++ 编程语言是比较容易的。开发 C++ 技巧，学习许多人已在使用的语言，在自己的编程工具箱中就会多一种功能强大的新工具。

为什么使用 ANSI/ISO 标准

自从 1989 年以来，C++ 就已成为目前正在进行的标准化进程的主题。这是 ANSI/ISO 小组的工作，其目的是为 C++ 语言开发一种世界标准。1997 年 11 月 14 日，该小组宣布已完成了 Final Draft International Standard (ISO/IEC FDIS 14882) (要了解更多细节，可访问站点 <http://www.maths.warwick.ac.uk/cpp/pub/>)，就剩下筹划指导委员会的最后(正式)投票，以及文档的最后完稿了——所有的这些工作本来计划在 1998 年完成。

标准的开发是一个相当困难的过程。1995 年 3 月对草案标准进行的第一次投票，使 C++ 语言发生了一些重大的变化。最终，Final Draft International Standard (国际标准最终草案) 标志着目前最重要的、使用非常广泛的编程语言 C++ 已经成熟，同时也表明许多 C++ 图书是不完整的。这也是编写本书的一个原因。

上面所谈的都是理论，该标准在实践中应如何应用？现在，应用于各种平台的 C++ 编译器随处可见，但这些编译器完全支持标准吗？一些编译器在遵循标准方面做得比另外一些编译器要好。多数编译器都使用 1995 的草案标准，所以人们希望在新的一年里能看到更新后的版本。

但是，显然读者希望编译和使用后面章节中的源代码。本书将介绍该标准，但只要该标准的某个特性并没有得到大量的应用，本书就会在相应的章节中指出。其后的代码将遵循该标准，幸好这种情况并不是很多。

练习

每一章的最后都有一些习题，用于测试读者对该章及以前章节的内容的掌握情况。所有习

题的答案都可以在 Web 站点上获得。如果读者有特别好的解决方案，可以给我们发送过来，我们将把它放在站点上。

读者反馈

我们尽可能使本书所述内容准确、通俗易懂。但问题是本书真的适合读者吗？请读者把想法告诉我们，可以通过 Wrox 出版社的 feedback@wrox.com 跟与我们联系。

源代码及更新

为了使本书物有所值，我们不再附赠一张光盘，而是把源代码、习题答案和本书的项目示例放在 Web 站点上：

<http://www.wrox.com/>

<http://www.wrox.co.uk/>

代码也可以通过 FTP 获得：

<ftp://ftp.wrox.com>

<ftp://ftp.wrox.co.uk>

错误和更正

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，请给我们发来反馈信息。

在读者访问如下 Web 站点时，请查看勘误表，该表专门用于修正本书中的小错误，或者提供解决问题的新方法及其解决方案。勘误表适用于所有的图书，读者可以下载它们，或参与到本书后续版本的改进中，上传更正后的内容，或给出新的解决方案。

<http://www.wrox.com>

<http://www.wrox.co.uk>

目 录

第 1 章 基本概念	1
1.1 编程语言	1
1.1.1 解释性的程序和编译性的程序的执行过程	1
1.1.2 库	2
1.2 为什么 C++ 是一种强大的语言	2
1.3 一个简单的 C++ 程序	3
1.4 名称	4
1.5 关键字	6
1.6 C++ 语句和语句块	6
1.7 程序结构	7
1.8 从源文件中创建可执行文件	8
1.8.1 编译	9
1.8.2 链接	9
1.9 字符和字符集	10
1.9.1 字符编码	11
1.9.2 转义序列	11
1.9.3 空白	13
1.10 程序的注释	14
1.11 标准库	15
1.12 用 C++ 编程	16
1.13 本章小结	
1.14 练习	
第 2 章 基本数据类型和计算	19
2.1 数据和数据类型	19
2.2 进行简单的计算	19
2.2.1 整型字面量	20
2.2.2 二进制数	21
2.2.3 十六进制数	22
2.2.4 负的二进制数	23
2.2.5 十六进制的整型字面量	24
2.2.6 八进制的整型字面量	25
2.2.7 整数的算术运算	25

2.3 使用变量	30
2.4 整型变量	31
2.4.1 整型变量类型	33
2.4.2 内存中的整型变量	35
2.5 赋值运算符	36
2.5.1 多次赋值	37
2.5.2 修改变量的值	37
2.6 整数的递增和递减	39
2.7 const 关键字	40
2.8 浮点数	42
2.8.1 浮点数的数据类型	43
2.8.2 使用浮点数值	45
2.9 使用字符	48
2.10 初始值的函数表示法	51
2.11 本章小结	51
2.12 练习	52
第3章 处理基本数据类型	53
3.1 混合的表达式	53
3.1.1 赋值和不同的类型	54
3.1.2 显式强制转换	55
3.1.3 老式的强制转换	58
3.2 确定类型	58
3.3 按位运算符	62
3.3.1 移位运算符	62
3.3.2 位模式下的逻辑运算	64
3.4 枚举数据类型	73
3.4.1 匿名枚举	74
3.4.2 在整型和枚举类型之间强制转换	75
3.5 数据类型的同义词	77
3.6 变量的生存期	78
3.6.1 自动变量	78
3.6.2 定位变量的声明	81
3.6.3 全局变量	81
3.6.4 静态变量	84
3.7 特殊的类型修饰符	84
3.8 声明外部变量	84
3.9 优先级和相关性	85

3.10 本章小结	86
3.11 练习	86
第 4 章 选择和决策	87
4.1 比较数据值	87
4.1.1 应用比较运算符	88
4.1.2 比较浮点数值	90
4.2 if 语句	90
4.3 if-else 语句	97
4.4 逻辑运算符	102
4.4.1 逻辑与运算符	103
4.4.2 逻辑或运算符	103
4.4.3 逻辑非运算符	104
4.5 条件运算符	107
4.6 switch 语句	109
4.7 无条件分支	114
4.8 决策语句块和变量作用域	114
4.9 本章小结	116
4.10 练习	116
第 5 章 循环	117
5.1 理解循环	117
5.2 while 循环	118
5.3 do-while 循环	120
5.4 for 循环	123
5.4.1 循环和变量作用域	126
5.4.2 用浮点数值控制 for 循环	128
5.4.3 使用更复杂的循环控制表达式	131
5.5 嵌套的循环	135
5.6 跳过循环迭代	139
5.7 循环的中断	141
5.8 本章小结	147
5.9 练习	148
第 6 章 数组和字符串	149
6.1 数据数组	149
6.1.1 使用数组	149
6.1.2 初始化数组	154
6.1.3 字符数组	158

6.2	多维数组	161
6.2.1	初始化多维数组	163
6.2.2	多维字符数组	166
6.3	string 类型	168
6.3.1	声明 string 对象	169
6.3.2	使用 string 对象	170
6.3.3	访问字符串中的字符	173
6.3.4	访问子字符串	175
6.3.5	比较字符串	175
6.3.6	搜索字符串	181
6.3.7	修改字符串	189
6.4	string 类型的数组	194
6.5	本章小结	195
6.6	练习	196
第 7 章	指针	197
7.1	什么是指针	197
7.2	指针的声明	198
7.3	指针的初始化	203
7.4	常量指针和指向常量的指针	214
7.5	指针和数组	216
7.5.1	指针的算术运算	216
7.5.2	使用数组名的指针表示法	218
7.5.3	对多维数组使用指针	222
7.5.4	C 样式字符串的操作	225
7.6	动态内存分配	226
7.6.1	自由存储区	227
7.6.2	运算符 new 和 delete	227
7.6.3	数组的动态内存分配	228
7.6.4	动态内存分配的危险	231
7.6.5	转换指针	237
7.7	本章小结	237
7.8	练习	238
第 8 章	使用函数编程	239
8.1	程序的分解	239
8.2	理解函数	240
8.2.1	定义函数	241
8.2.2	函数的声明	245

8.3	给函数传送参数	247
8.3.1	按值传送机制	247
8.3.2	按引用传送机制	256
8.3.3	main()的参数	260
8.4	默认的参数值	261
8.5	从函数中返回值	265
8.5.1	返回一个指针	265
8.5.2	返回一个引用	270
8.5.3	从函数中返回新变量	270
8.6	内联函数	270
8.7	静态变量	271
8.8	本章小结	274
8.9	练习	274
第 9 章	函数	276
9.1	函数的重载	276
9.1.1	函数的签名	276
9.1.2	重载和指针参数	279
9.1.3	重载和引用参数	279
9.1.4	重载和 const 参数	281
9.1.5	重载和默认参数值	282
9.2	函数模板	283
9.2.1	显式指定模板参数	286
9.2.2	模板的说明	287
9.2.3	函数模板和重载	290
9.2.4	带有多个参数的模板	291
9.3	函数指针	292
9.3.1	声明函数指针	293
9.3.2	把函数作为参数传送	296
9.3.3	函数指针的数组	298
9.4	递归	299
9.5	本章小结	307
9.6	练习	308
第 10 章	程序文件和预处理器	309
10.1	使用程序文件	309
10.2	命名空间	315
10.2.1	全局命名空间	316
10.2.2	声明命名空间	317

10.2.3	隐式地限定名称	320
10.2.4	函数和命名空间	322
10.2.5	函数模板和命名空间	326
10.2.6	扩展命名空间	328
10.2.7	未指定名称的命名空间	332
10.2.8	命名空间的别名	332
10.2.9	嵌套的命名空间	333
10.3	预处理器	334
10.3.1	在程序中包含头文件	335
10.3.2	程序中的置换	336
10.3.3	宏置换	338
10.3.4	放在多行代码中的预处理器指令	340
10.3.5	把字符串作为宏参数	340
10.3.6	在宏表达式中连接参数	342
10.4	逻辑预处理器指令	342
10.4.1	逻辑#if 指令	342
10.4.2	测试特定值的指令	345
10.4.3	多个代码选择块	345
10.4.4	标准的预处理器宏	346
10.4.5	#error 和#pragma 指令	347
10.5	调试方法	348
10.5.1	集成调试器	348
10.5.2	调试中的预处理器	349
10.5.3	使用 assert 宏	355
10.6	本章小结	357
10.7	练习	358
第 11 章	创建自己的数据类型	359
11.1	对象的概念	359
11.2	C++中的结构	360
11.2.1	理解结构	360
11.2.2	定义结构类型	361
11.2.3	初始化结构变量	363
11.2.4	访问结构变量的成员	364
11.2.5	对结构使用指针	370
11.3	联合	373
11.3.1	声明联合	374
11.3.2	匿名联合	376

11.4	更复杂的结构	377
11.5	本章小结	384
11.6	练习	384
第 12 章	类	385
12.1	类和面向对象编程	385
12.1.1	封装	386
12.1.2	继承	387
12.1.3	多态性	388
12.1.4	术语	389
12.2	定义类	390
12.3	构造函数	393
12.3.1	把构造函数的定义放在类的外部	395
12.3.2	默认的构造函数	397
12.3.3	默认的初始化值	401
12.3.4	在构造函数中使用初始化列表	402
12.3.5	使用 <code>explicit</code> 关键字	402
12.4	类的私有成员	404
12.4.1	访问私有类成员	408
12.4.2	默认的副本构造函数	410
12.5	友元	411
12.5.1	类的友元函数	412
12.5.2	友元类	415
12.6	<code>this</code> 指针	415
12.7	<code>const</code> 对象和 <code>const</code> 成员函数	419
12.7.1	类中的 <code>mutable</code> 数据成员	421
12.7.2	常量的强制转换	422
12.8	类的对象数组	422
12.9	类对象的大小	425
12.10	类的静态成员	427
12.10.1	类的静态数据成员	428
12.10.2	类的静态成员函数	433
12.11	本章小结	435
12.12	练习	435
第 13 章	类的操作	437
13.1	类对象的指针和引用	437
13.2	指针作为数据成员	437
13.2.1	定义 <code>Package</code> 类	439

13.2.2	定义 TruckLoad 类	442
13.2.3	实现 TruckLoad 类	443
13.3	控制对类的访问	452
13.4	副本构造函数的重要性	455
13.5	对象内部的动态内存分配	461
13.5.1	析构函数	462
13.5.2	默认的析构函数	462
13.5.3	实现析构函数	465
13.6	类的引用	466
13.7	本章小结	469
13.8	练习	469
第 14 章	运算符重载	471
14.1	为自己的类实现运算符	471
14.1.1	运算符重载	471
14.1.2	可以重载的运算符	472
14.1.3	实现重载运算符	472
14.1.4	全局运算符函数	476
14.1.5	提供对运算符的全部支持	477
14.1.6	运算符函数术语	481
14.1.7	重载赋值运算符	481
14.1.8	重载算术运算符	489
14.1.9	重载下标运算符	494
14.1.10	重载类型转换	502
14.1.11	重载递增和递减运算符	504
14.1.12	智能指针	504
14.1.13	重载运算符 new 和 delete	511
14.2	本章小结	512
14.3	练习	512
第 15 章	继承	514
15.1	类和面向对象编程	514
15.2	类的继承	516
15.2.1	继承和聚合	516
15.2.2	从基类中派生新类	517
15.3	继承下的访问控制	521
15.4	把类的成员声明为 protected	524
15.5	派生类成员的访问级别	526
15.6	派生类中的构造函数操作	530

15.7	继承中的析构函数	538
15.8	重复的成员名	540
15.9	多重继承	542
15.9.1	多个基类	542
15.9.2	继承成员的模糊性	543
15.9.3	重复的继承	550
15.9.4	虚基类	551
15.10	在相关的类类型之间转换	552
15.11	本章小结	553
15.12	练习	553
第 16 章	虚函数和多态性	555
16.1	理解多态性	555
16.1.1	使用基类指针	555
16.1.2	调用继承的函数	557
16.1.3	虚函数	561
16.1.4	虚函数中的默认参数值	569
16.1.5	通过引用来调用虚函数	570
16.1.6	调用虚函数的基类版本	572
16.1.7	在指针和类对象之间转换	572
16.1.8	动态强制转换	574
16.2	多态性的成本	576
16.3	纯虚函数	577
16.3.1	抽象类	578
16.3.2	间接的抽象基类	581
16.4	通过指针释放对象	585
16.5	在运行期间标识类型	587
16.6	类成员的指针	589
16.6.1	数据成员指针	589
16.6.2	成员函数指针	593
16.7	本章小结	597
16.8	练习	598
第 17 章	程序错误和异常处理	599
17.1	处理错误	599
17.2	理解异常	600
17.2.1	抛出异常	600
17.2.2	导致抛出异常的代码	606
17.2.3	嵌套的 try 块	607

17.3	用类对象作为异常	611
17.3.1	匹配 Catch 处理程序和异常	613
17.3.2	用基类处理程序捕获派生类异常	616
17.3.3	重新抛出异常	619
17.3.4	捕获所有的异常	623
17.4	抛出异常的函数	625
17.4.1	函数 try 块	625
17.4.2	在构造函数中抛出异常	628
17.4.3	异常和析构函数	629
17.5	标准库异常	629
17.5.1	标准库异常类	630
17.5.2	使用标准异常	631
17.6	本章小结	632
17.7	练习	632
第 18 章	类模板	634
18.1	理解类模板	634
18.2	定义类模板	635
18.2.1	模板参数	636
18.2.2	简单的类模板	636
18.2.3	创建类模板的实例	641
18.2.4	类模板的静态成员	649
18.2.5	非类型的类模板参数	650
18.2.6	非类型参数示例	651
18.2.7	默认的模板参数值	662
18.3	模板的显式实例化	662
18.4	类模板的友元	663
18.5	特殊情形	664
18.6	带有嵌套类的类模板	666
18.7	更高级的类模板	676
18.8	本章小结	676
18.9	练习	677
第 19 章	输入输出操作	678
19.1	C++中的输入输出	678
19.2	流类	680
19.2.1	标准流	681
19.2.2	流的插入和提取操作	681
19.2.3	流操纵程序	683

19.3	文件流	686
19.3.1	写入文件	686
19.3.2	读取文件	689
19.3.3	设置文件打开模式	691
19.4	未格式化的流操作	701
19.4.1	未格式化的流输入函数	701
19.4.2	未格式化的流输出函数	703
19.5	流输入输出中的错误	703
19.6	使用二进制模式流操作	705
19.7	流上的读写操作	714
19.8	字符串流	723
19.9	对象和流	724
19.9.1	重载类对象的插入运算符	724
19.9.2	重载类对象的提取运算符	727
19.9.3	流中更复杂的对象	730
19.10	本章小结	742
19.11	练习	743
第 20 章	标准模板库	744
20.1	STL 架构简介	744
20.2	使用 vector 容器	746
20.2.1	vector 容器的基本操作	748
20.2.2	使用 vector 容器进行数组操作	754
20.2.3	istream_iterator 迭代器	759
20.3	理解迭代器	762
20.3.1	给算法传送迭代器	764
20.3.2	模拟 istream_operator 和 ostream_operator	766
20.3.3	迭代器的要求	767
20.3.4	修订后的 Integer 迭代器	769
20.3.5	输入和输出迭代器	776
20.4	存储空间管理一瞥	777
20.4.1	Account 帮助类	778
20.4.2	STL 元素需求和理想的 Bean	780
20.4.3	确定 vector 的大小和 vector 存储空间的管理	783
20.5	list 容器	789
20.6	关联 map 容器	795
20.7	性能和规范	801
20.8	本章小结	803