

对一个即将出门旅行的人来说，最需要的是
一张内容详尽、生动、实用的旅行地图。

对即将进入神秘、美丽的 C++ 世界的你，最
需要的是这本 C++ 世界地图册。

本书将带领你畅游整个 C++ 世界。还等什么，
让我们出发吧！

我的第一本

C++ 书



陈良乔 著

我的第一本 C++ 书

陈良乔 著

华中科技大学出版社

中国·武汉

内 容 简 介

虽然 C++ 语言纷繁复杂的语法规则让很多学习者望而却步,但是,在本书中,你会发现 C++ 语言的学习也可以如此轻松。本书没有孔乙己式地去深究 C++ 语言的语法细节,也没有重点地去介绍各种高深的 C++ 编程技巧,而是本着简单实用的原则,通俗易懂地向你介绍 C++ 中最重要、最实用的知识。看完本书,你会发现用逻辑控制语句可以把 C++ 语句串珠成链;用函数实际上就是把程序装进一个箱子;当 C++ 语言爱上面向对象思想就有了类与对象;算法就像体育老师,给我们带来一堂别开生面的算法体育课。本书不再是一本枯燥乏味的 C++ 语法介绍书,而是一本通俗易懂的 C++ 故事书,可以让我们在讲故事中轻松学会 C++ 语言。

图书在版编目(CIP)数据

我的第一本 C++ 书/陈良乔 著. —武汉:华中科技大学出版社,2011.5

ISBN 978-7-5609-6995-4

I. 我… II. 陈… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2011)第 044235 号

我的第一本 C++ 书

陈良乔 著

策划编辑:陈禹成

责任编辑:陈元玉

封面设计:潘 群

责任校对:周 娟

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:华中科技大学惠友文印中心

印 刷:湖北新华印务有限公司

开 本:787mm×1092mm 1/16

印 张:24.5

字 数:608 千字

版 次:2011 年 5 月第 1 版第 1 次印刷

定 价:58.80 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

目录

第 1 篇 叩开 C++ 世界的大门	1
第 1 章 C++ 世界地图	3
1.1 C++ 是什么	3
1.2 C++ 的前世今生	3
1.2.1 从 B 到 C	4
1.2.2 从 C 到 C++	4
1.2.3 从 C++ 到 .NET Framework 的 CLI	6
1.2.4 最新标准 C++0x 让 C++ 重新焕发活力	8
1.2.5 C++ 和 C# 不得不说的那点儿事儿	8
1.2.6 C++ 世界的五大子语言	9
1.3 C++ 世界版图	10
1.3.1 Windows 系统下的 C++ 开发	10
1.3.2 Linux 系统下的 C++ 开发	11
1.3.3 嵌入式系统下的 C++ 开发	11
1.4 如何学好 C++	12
1.4.1 将自然语言转换为 C++ 程序设计语言	12
1.4.2 “多读多写”是学好 C++ 的不二法门	14
1.4.3 和 Google 做朋友	14
第 2 章 与 C++ 第一次亲密接触	17
2.1 一个 C++ 程序的自白	17
2.1.1 用 Visual Studio 创建 C++ 程序	17
2.1.2 以手工方式创建 C++ 程序	20
2.1.3 C++ 程序=预编译指令+程序代码+注释	21
2.1.4 编译器和链接器	25
2.1.5 C++ 程序的执行过程	26
2.1.6 程序的两大任务：描述数据与处理数据	27

2.2	基本输入/输出流.....	28
2.2.1	标准的输入和输出对象.....	29
2.2.2	输出格式控制.....	31
2.2.3	读/写文件.....	32
2.3	最常用的开发环境 Visual Studio.....	34
2.3.1	Visual C++的常用菜单.....	35
2.3.2	Visual C++的常用视图.....	44
2.4	C++世界旅行必备的物品.....	46
2.4.1	编程助手 Visual Assist.....	46
2.4.2	代码配置管理工具 Visual Source Safe.....	48
2.4.3	CodeProject 和 CodeGuru.....	48
2.4.4	C++百科全书 MSDN.....	48
第 2 篇 欢迎来到 C++世界.....		51
第 3 章 C++世界众生相.....		53
3.1	C++中的数据类型.....	53
3.2	变量和常量.....	54
3.2.1	声明变量.....	55
3.2.2	给变量取个好名字.....	55
3.2.3	变量初始化.....	57
3.2.4	常量.....	57
3.2.5	用宏与 const 关键字定义常量.....	59
3.3	数值类型.....	62
3.3.1	整型数值类型.....	62
3.3.2	浮点型数值类型.....	63
3.4	布尔类型.....	64
3.5	字符串类型.....	65
3.5.1	字符类型.....	65
3.5.2	字符串类型.....	66
3.6	数组.....	67
3.6.1	数组的声明与初始化.....	68
3.6.2	数组的使用.....	69
3.7	枚举类型.....	71

3.8	用结构体类型描述复杂的事物	73
3.8.1	结构体的定义	73
3.8.2	结构体的使用	74
3.9	指向内存位置的指针	76
3.9.1	指针就是表示内存地址的数据类型	76
3.9.2	指针变量的定义	77
3.9.3	指针的赋值和使用	78
第 4 章	将语句编织成程序	81
4.1	用运算符对数据进行运算	81
4.1.1	用表达式表达设计意图	82
4.1.2	算术运算符	82
4.1.3	赋值操作符	84
4.1.4	关系运算符	84
4.1.5	逻辑运算符	86
4.1.6	运算符之间的优先顺序	87
4.1.7	将表达式组织成语句	89
4.2	条件选择语句	90
4.2.1	if 语句	90
4.2.2	并列选择的 switch 语句	93
4.3	循环控制语句	97
4.3.1	while 循环	97
4.3.2	do...while 循环	99
4.3.3	for 循环	100
4.3.4	循环控制: break 和 continue	102
4.4	从语句到程序	104
4.4.1	程序是控制语句串联起来的语句	104
4.4.2	豪华工资统计程序	106
第 5 章	用函数封装程序功能	109
5.1	函数就是一个大“箱子”	109
5.1.1	函数的声明和定义	110
5.1.2	函数调用机制	113
5.1.3	函数的声明与函数调用	117
5.1.4	函数参数的传递	119

5.1.5	函数的返回值	121
5.2	内联函数	123
5.2.1	用体积换速度的内联函数	123
5.2.2	内联函数的使用规则	124
5.3	重载函数	125
5.3.1	重载函数的声明	125
5.3.2	重载函数的解析	128
5.4	函数设计的基本规则	129
5.4.1	函数声明的设计规则	129
5.4.2	函数体的设计规则	131
第 6 章	当 C++ 爱上面向对象	135
6.1	从结构化设计到面向对象程序设计	135
6.1.1	“自顶向下，逐步求精”的结构化程序设计	136
6.1.2	面向对象程序设计	137
6.1.3	面向对象的三座基石：封装、继承与多态	138
6.2	类：当 C++ 爱上面向对象	142
6.2.1	类的声明和定义	142
6.2.2	使用类创建对象	146
6.2.3	构造函数和析构函数	148
6.2.4	拷贝构造函数	152
6.2.5	操作符重载	155
6.2.6	类成员的访问控制	157
6.2.7	在友元中访问类的隐藏信息	160
6.3	类如何面向对象	162
6.3.1	用类机制实现封装	163
6.3.2	用基类和派生类实现继承	164
6.3.3	用虚函数实现多态	171
6.4	实战面向对象：工资管理系统	175
6.4.1	从问题描述中发现对象	175
6.4.2	分析对象的属性和行为	176
6.4.3	实现类的属性和行为	177
6.5	高手是这样炼成的	183
6.5.1	C++ 类对象的内存模型	183

6.5.2 指向自身的 this 指针.....	185
第 7 章 C++世界的奇人异事.....	189
7.1 一切指针都是纸老虎：彻底理解指针.....	189
7.1.1 指针的运算.....	189
7.1.2 灵活的 void 类型和 void 类型指针.....	192
7.1.3 指向指针的指针.....	194
7.1.4 指针在函数中的应用.....	195
7.1.5 引用.....	198
7.2 程序中的异常处理.....	203
7.2.1 异常处理.....	203
7.2.2 异常的函数接口声明.....	206
7.2.3 合理使用异常处理.....	207
7.3 编写更复杂的 C++程序.....	208
7.3.1 源文件和头文件.....	208
7.3.2 名字空间.....	210
7.3.3 作用域与可见性.....	214
7.3.4 编译预处理.....	218
7.4 高手是这样炼成的.....	220
7.4.1 用宏定义化繁为简.....	220
7.4.2 用 typedef 定义类型的别名.....	221
7.4.3 用 const 保护数据.....	223
第 3 篇 攀登 C++世界的高峰.....	227
第 8 章 用 STL 优雅你的程序.....	229
8.1 跟 STL 做朋友.....	230
8.1.1 算法 + 容器 + 迭代器 = STL.....	230
8.1.2 在程序中使用 STL.....	231
8.1.3 STL 到底好在哪里.....	233
8.2 用模板实现通用算法.....	234
8.2.1 函数模板.....	235
8.2.2 类模板.....	238
8.2.3 模板的实例化.....	240
8.2.4 用模板实现通用算法.....	241

第 9 章	STL 中的容器管理数据	245
9.1	容器就是 STL 中的瓶瓶罐罐	245
9.1.1	操作容器中的数据元素	247
9.1.2	使用迭代器访问容器中的数据元素	247
9.1.3	容器的使用说明书	249
9.1.4	如何选择合适的容器	252
9.2	vector 容器是数组的最佳替代者	253
9.2.1	创建并初始化 vector 对象	253
9.2.2	vector 容器的操作	256
9.2.3	访问 vector 容器中的数据	257
9.3	可以保存键值对的 map 容器	258
9.3.1	创建并初始化 map 容器	258
9.3.2	将数据保存到 map 容器中	259
9.3.3	根据键找到对应的值	260
第 10 章	用 STL 中的通用算法处理数据	263
10.1	STL 算法中的“四大帮派”	263
10.2	容器元素的查找与遍历	264
10.2.1	用 for_each() 算法遍历容器中的数据元素	264
10.2.2	用 find() 和 find_if() 算法实现线性查找	266
10.3	容器元素的复制与变换	270
10.3.1	复制容器元素: copy() 算法	270
10.3.2	合并容器元素: merge() 算法	272
10.3.3	变换容器元素: transform 函数	274
10.4	容器元素的排序	276
10.4.1	使用 sort() 算法对容器中的数据进行排序	276
10.4.2	对排序的规则进行自定义	279
10.5	实战 STL 算法	282
10.5.1	“算法”老师带来的一堂别开生面的体育课	282
10.5.2	删除容器中的冗余元素	284
第 11 章	函数指针、函数对象与 Lambda 表达式	287
11.1	函数指针	287
11.1.1	函数指针的声明与赋值	287
11.1.2	用函数指针调用函数	289

11.1.3	用函数指针实现回调函数.....	291
11.1.4	将函数指针应用到 STL 算法中.....	293
11.2	函数对象.....	295
11.2.1	定义一个函数对象.....	295
11.2.2	利用函数对象记住状态数据.....	297
11.3	用 Lambda 表达式编写更简单的函数.....	299
11.3.1	最简单直接的函数表达形式：Lambda 表达式.....	299
11.3.2	Lambda 表达式的语法规则.....	301
11.3.3	Lambda 表达式的复用.....	302
第 12 章	C++世界的几件新鲜事.....	305
12.1	用右值引用榨干 C++的性能.....	305
12.1.1	什么是右值.....	305
12.1.2	右值引用在函数返回值上的应用.....	306
12.1.3	STL 算法中被浪费的右值.....	307
12.1.4	右值引用如何提高性能.....	310
12.2	智能指针 shared_ptr.....	312
12.2.1	C++的内存管理.....	312
12.2.2	用聪明的 shared_ptr 解决内存管理问题.....	313
12.2.3	智能指针的应用场景.....	315
12.2.4	shared_ptr 的使用.....	316
12.2.5	shared_ptr 与标准库容器.....	321
12.2.6	对 shared_ptr 进行自定义.....	323
12.3	用 PPL 进行多线程开发.....	326
12.3.1	多核给程序设计带来的挑战.....	326
12.3.2	PPL 带来免费的午餐.....	327
12.3.3	PPL 中的并行算法.....	329
12.3.4	PPL 中的并行任务.....	331
12.3.5	PPL 中的并行对象和并行容器.....	334
12.3.6	PPL 之外的另一种选择：OpenMP.....	340
第 13 章	找工作就靠它了.....	343
13.1	打好基础.....	343
13.1.1	基本概念.....	343
13.1.2	函数.....	346

13.1.3 面向对象思想	348
13.1.4 类与对象	349
13.1.5 STL	360
13.2 积累经验	362
13.3 考查智力	364
附录 A 接下来该读什么书	367
A.1 开山鼻祖：《C++程序设计语言》	367
A.2 初学者必看：《C++ Primer 中文版（第4版）》	368
A.3 百科全书：《代码大全，第2版》	368
A.4 内功秘籍：《Effective C++（Third Edition）》	370
A.5 经验很重要：《C++编程规范》	370
后记	373

前言

当你拿起本书翻看到这一页的时候，是否在寻找一本既简单实用又通俗易懂的 C++ 参考书？没错，这本书正是你要寻找的最佳 C++ 参考书。

有了这本书的帮助，你的 C++ 学习过程将会是一次愉快的 C++ 世界之旅。用 C++ 编程就是用 C++ 编程语言描述和表达周围现实世界。按照描述和表达现实世界的需要，本书分别介绍了 C++ 语言中最重要的内容，包括基本数据类型、逻辑控制语句、函数、面向对象思想、类与对象，以及标准模板库等相关的知识。学习基本数据类型之后，你会知道如何使用 `int`、`double` 和 `string` 等数据类型来描述现实世界中的数据；学习逻辑控制语句之后，你会知道如何使用 `if...else` 来控制程序逻辑；学习函数之后，你会知道如何用函数来表达完整的算法；而面向对象思想可以帮助我们抽象现实世界；类与对象则将抽象的结果在程序中表达出来；最后的标准模板库，通过提供通用容器来对数据进行管理，通过提供通用算法来对数据进行处理，从而使程序更加简单而优雅。这些内容，都是 C++ 语言中最基本而又最实用的部分，通过这些内容的学习，你完全可以将 C++ 作为自己的语言来描述和表达现实世界。

这么多内容，学习起来困难吗？虽然 C++ 语言纷繁复杂的语法规则让很多学习者望而却步，但是，在本书中，你会发现 C++ 语言的学习也可以如此轻松。本书没有孔乙己式地去深究 C++ 语言的语法细节，也没有重点地去介绍各种高深的 C++ 编程技巧，而是本着简单实用的原则，通俗易懂地向你介绍 C++ 中最重要、最实用的知识。看完本书，你会发现用逻辑控制语句可以把 C++ 语句串珠成链；用函数实际上就是把程序装进一个箱子；当 C++ 语言爱上面向对象思想就有了类与对象；算法就像体育老师，给我们带来一堂别开生面的算法体育课。本书不再是一本枯燥乏味的 C++ 语法介绍书，而是一本通俗易懂的 C++ 故事书，可以让我们在讲故事中轻松学会 C++ 语言。

本书不仅仅是一本入门参考书，它更是一本关于 C++ 编程经验总结的书。本人使用 C++ 编程已有 10 多年时间，同时多年担任微软最有价值专家（Microsoft Most Valuable Professional, MVP），接触到很多来自实践的问题，也积累了丰富的实践经验。我的这些宝贵实践经验都融入了本书中。通过本书，你不仅可以获得关于 C++ 语言的知识，更重要的，你可以获得这些宝贵的实践经验。例如，表示浮点数的 `float` 和 `double`，我们到底该如何选择？为什么 `const` 关键字这么重要？指针和引用的区别到底在哪里？这些都是来自实践的问题，

在你读完这本书后，你将获得完美的答案，这些答案会帮助你从一个 C++ 新手成长为一个经验丰富的 C++ 开发人员。如果你已经在进行 C++ 程序开发，也可以通过这些经验的积累，让自己的 C++ 编程技能更上一层楼。

最后，当你在 C++ 学习之旅中累了、倦了，或者遇到问题的时候，不妨来我的“有{問}客栈”坐坐，链接地址：<http://imcc.blogbus.com/>。在这里，我可以为你答疑解惑，让你的 C++ 学习之旅更加轻松惬意。

还等什么呢？即刻拿起这本书，开始我们的 C++ 学习之旅吧！

致谢

本书从最初提出设想到最终成书经历了大约四年时间，我希望能是四年磨一剑，能给大家奉上一本好书，一本有价值的书。

在这四年的写作过程中，我获得了很多人的帮助，最终才有了大家手中的这本书。在这里，我要感谢父母对我的默默支持；感谢亲爱的贾玮，你的支持、鼓励和期许是我的动力和灵感的源泉；感谢为本书作序的康雁老师，我的好老师；感谢周静姐姐、唐总和兰姐给予我的教诲；感谢好朋友丁春利、马洪旭在我最困难的时候给予我的无私帮助；感谢 Elmar Driesch 先生、Andreas Muench 先生及 Thorsten Thomsen 先生给予我的指导；感谢本书的编辑，从最初的陈禹成编辑到后来的徐定翔、陈元玉编辑，是你们成就了这本书；感谢所有关心我的朋友，我爱你们！

陈良乔

2011 年 2 月于西安

序言

在多年的教学实践中，我深深感到 C++ 语言的灵活和高效，以及 C++ 语言中的面向对象特性能够带给软件开发者无尽想象的空间，同时也深深感到教授 C++ 语言过程中面临的困难和挑战。尽管目前有关 C++ 语言的书籍很多，但学习 C++ 语言仍然让大多数初学者心存畏惧。本书作者结合自己学习和使用 C++ 语言的经验和感悟，尝试用轻松幽默的语言，借助人们熟知的生活概念，形象生动地讲述 C++ 语言。本书对那些渴望掌握 C++ 语言而又心存畏惧的初学者，无疑是一个很好的选择。

作者与我的团队一同快速成长，相信在这个成长的过程中作者也如我一样，经历了从在懵懂中模仿使用 C++ 语言到喜爱 C++ 语言的指针的灵活及面向对象的设计思想。尽管近年来新的计算机语言不断出现，但 C++ 语言因同时具有最贴近计算机数值操作方式的指针操作和最贴近人类社会行为模式的面向对象的特性，而具有其他程序语言无法比拟的强大生命力。可以说，指针和面向对象的特性是 C++ 语言的灵魂，同时也是学习和使用 C++ 的难点。期望那些有所感悟的同行，参与到探寻有效学习 C++ 语言的方法中来。

本书全面阐述了 C++ 语言的基本概念和技术，并且结合了实例进行讲解，另外，还给出了一些公司的典型笔试题，更加突显出其实用性。本书的语言轻松、幽默，经过作者的深入思考和策划使得学习 C++ 不再枯燥。本书的美中不足之处在于讲解还可以进一步深入、个别观点和类比可以进一步完善。如果读者能够结合书后附录中给出的参考书，会收到更好的学习效果。

康 雁

2011 年 3 月 26 日于沈阳

第1篇

叩开〇++世界的大门

C++世界地图

对一个即将到陌生地方去旅行的人来说，什么是最重要的和必需的？没错，是一张内容丰富详尽、生动有趣的旅行地图。借助这张地图，就知道在什么地方停车吃饭、在什么地方打尖住店。即将进入陌生 C++ 世界的各位旅行者对 C++ 世界有太多的问题和疑惑。C++ 是什么？C++ 是怎么来的？C++ 能做什么？如何学好 C++？

面对这些问题，同样需要一张 C++ 世界地图。这张 C++ 世界地图可以解答这些问题和疑惑，让我们清晰地认识 C++ 世界。同时，我们可以通过这张 C++ 世界地图，了解 C++ 世界的整个面貌：有哪些好玩的地方，有哪些有趣的故事，有哪些有用的知识，有哪些危险而需要注意的地方。这张 C++ 世界地图，将带领我们畅游整个 C++ 世界。

还等什么，让我们出发吧！

1.1 C++是什么

C++ 是什么？这几乎是每个即将进入 C++ 世界的旅行者要问的第一个问题。面对这个问题，一千个人有一千种答案。有人说，C++ 是一种高级程序设计语言；有人说，C++ 是 C 语言的继承者；有人说，C++ 是一种面向对象的思维方式；有人说，C++ 很好很强大；更有人说，不要迷恋 C++，C++ 只是一个传说。可以说，这些答案都从某个侧面回答了“C++ 是什么”，但是又都不全面。

在这里，我不能、也不会给你所谓的完美答案。我能够做到的，就是给你描绘美好的 C++ 世界，讲述其中的各色公民、风俗见闻，为各位旅行者指引方向和路径。等各位完成这奇妙的 C++ 世界之旅之后，自然会有自己的答案。

从现在开始，寻找答案吧！

1.2 C++的前世今生

读史可以使人明智。

C++作为一门高级程序设计语言，可说是历史悠久，算得上是程序设计语言中的“老革命”了。了解 C++的发展历史，可以加深我们对这门语言的认识，了解 C++的本质内涵，了解 C++的文化，从而更好地学习和掌握这门语言。

传说，很久很久以前……

1.2.1 从 B 到 C

1967 年，著名的计算机科学家丹尼斯·里奇（Dennis Ritchie）进入美国 AT&T 的贝尔实验室工作。一开始，里奇和他的同事肯·汤普森（Ken Thompson）开始研究 DEC PDP-7 机器，但是他们发现在这个机器上写程序很困难，只能用汇编语言编程。于是汤普森设计了一种高级程序语言，命名为 B 语言。但是 B 语言本身设计的缺陷使汤普森在内存的限制面前一筹莫展。到了 1973 年，里奇对 B 语言进行改良，他赋予这门新语言强有力的系统控制方面的能力。新语言非常简洁、高效，里奇把它命名为 C 语言，意为 B 语言的下一代程序设计语言。

知道更多：B 语言从哪里来

C 语言来自 B 语言，那么 B 语言是不是来自 A 语言呢？B 语言之前并不存在 A 语言，之所以取名 B 语言，是为了纪念作者的妻子，他妻子名字的第一个字母是 B。

嗯，程序员中也有情圣啊！

1978 年，里奇和布朗一起出版了著名的《C Programming Language》一书，C 语言随后成为世界上应用最广泛的高级程序设计语言，这个版本的 C 语言被称为 K&R C。1989 年，C 语言被 ANSI 标准化（ANSI X3.159—1989）。标准化的目的是扩展 K&R C，加入新的特性。在 K&R C 发布后，又不断有人为 C 语言添加新特性，但 C 语言的标准在一段相当长的时间内都保持不变，直到 20 世纪 90 年代，标准才被更新，这就是 ISO 9899:1999（1999 年发布）。这个版本就是通常提及的 C99。ANSI 于 2000 年 3 月采用了这个新标准。

1.2.2 从 C 到 C++

语言的发展是一个逐步递进的过程。1979 年 4 月，同样来自贝尔实验室的本贾尼·斯特劳斯特卢普（Bjarne Stroustrup）博士与同事接受一项工作，尝试分析 UNIX 的内核。当时没有合适的工具能够有效地分析内核分布形成的网络流量，将内核模块化，所以斯特劳斯特卢普博士的工作进展很慢。同年 10 月，斯特劳斯特卢普博士设计了一个可以运行的预处理程序，称之为“Cpre”，它为 C 语言加上了类似 Simula 语言的类机制。在这个过程中，斯特劳斯特