

资深工程师编写，讲述最核心的编程问题，规避最可能遇到的误区！
配套视频、在线题库和讨论区，提供充分的示例源码！
主次分明，前后有序，循序渐进，娓娓道来；C/C++学习指南，让编程变得简单！



我的编程课堂

C/C++学习指南 (语法篇)

◎ 邵发 编著

视频讲解

在线题库

资源共享

在线互动

进阶教程



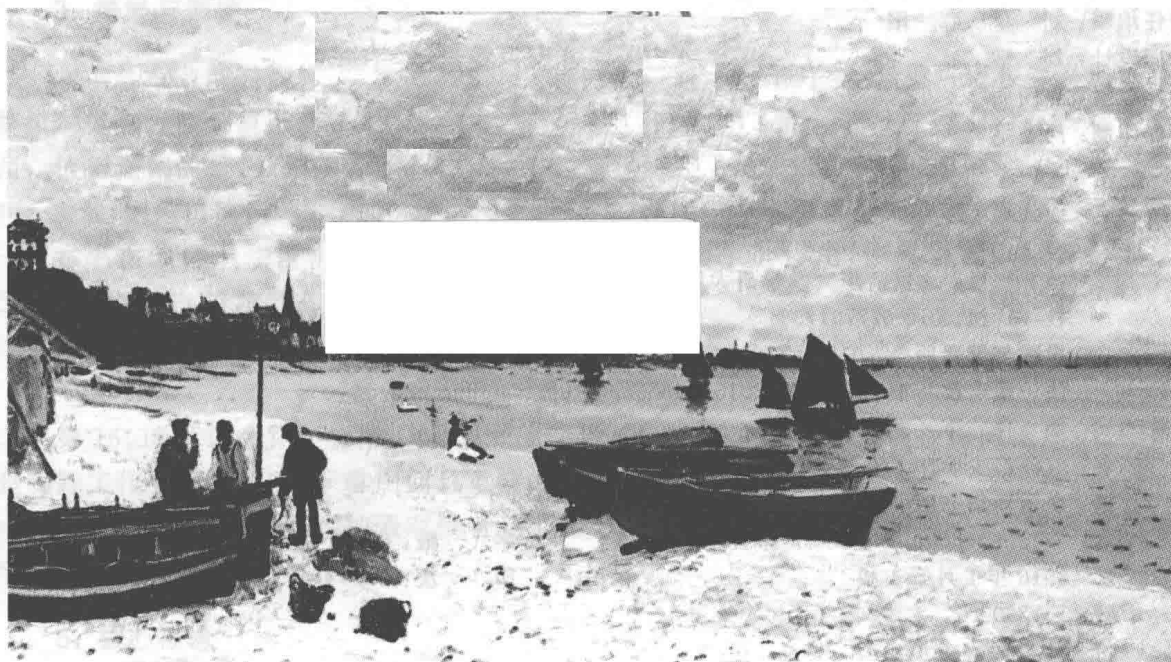
清华大学出版社

我的编程课堂

C/C++学习指南

(语法篇)

◎ 邵发 编著



清华大学出版社
北京

内 容 简 介

这是一部 C/C++ 语言的基础语法教程,面向零基础的初学者。C++ 是在 C 语言的基础上做了部分修改和扩展的程序设计语言。本书的前半部分(1~19 章)主要是 C 语言部分,而后半部分(20~31 章)则是 C++ 对 C 语言的扩展。

本书从实际工程需要出发,介绍一个 C/C++ 从业人员所必须掌握的全部语法、核心概念和基本技能,消除了学习考试与实际工作之间的差距。书中不仅介绍函数、指针、结构体、类等基本语法,也介绍了传值与传地址、字符串、深拷贝与浅拷贝、动态创建对象、安全地使用指针、面向对象编程思想等核心概念。同时,书中还讲解初级程序员所必须掌握的知识和技术,如单步调试、编码规范、ANSI 函数库、文件操作、标准模板库 STL 的使用。

本书配备免费的全套视频教程、在线题库及解析,可结合纸质教材同步使用,书中的近三百个示例的源码,以及所有相关资料均可在 <http://afanihao.cn> 上获取。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C/C++ 学习指南. 语法篇/邵发编著. --北京:清华大学出版社,2016

我的编程课堂

ISBN 978-7-302-41989-1

I. ①C… II. ①邵… III. ①C 语言—程序设计—指南 IV. ①TP312-62

中国版本图书馆 CIP 数据核字(2015)第 263789 号

责任编辑:刘 星 薛 阳

封面设计:刘 键

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.75 字 数:604 千字

版 次:2016 年 1 月第 1 版 印 次:2016 年 1 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

产品编号:067218-01

前言

作为一个有十余年软件开发背景的从业人员,作者在招聘面试中发现,现在的绝大多数应聘者往往缺少基本的语法基础和实际的动手能力,更不用说对核心概念的理解了。这似乎表明现有的教材或教育方法,与实际的工作要求之间存在着比较严重的脱节。因此,作者希望创建一套全新的、完整的、综合性的教学体系,让学员能够打下牢固的语法基础,建立对核心概念的正确认识,并掌握基本的动手能力。这套体系包含纸质教程、视频教程、在线题库、进阶教程 4 大要素。

本书是目标体系的一个组成部分,它也可以独立地使用。就这部纸质教材的特点来说,有以下几点。

1. 主次分明

C/C++ 的语法繁多,有些语法是过时的,有些语法很难但工作中并不常用。为了避免给初学者带来不必要的干扰,书中对非重点的章节采用星号标注,并显式地说明这个章节是不重要的、或者是过时的、或者是第一遍学习的时候可以跳过的。

2. 引例

在开始讲解每一章之前,通常会有一个引例,其作用是引起读者对某个背景问题的思考。引例描述了一个语法产生的背景,使读者自然而然地理解本章引入的新概念。

3. 铺垫与衔接

本书各章节前后逻辑相承,一环连接一环。读者只要按顺序学习,就可以很容易地从最简单的概念过渡到复杂的概念。最终的效果是,读者会感觉整个学习过程中没有太难的东西。值得强调的是,如果读者不按顺序学习,则学习难度会加倍。例如,如果读者不学习第 3、4、5 章,则难以读懂第 8、9、10 章。反之,如果按顺序学习则可以不知不觉地通过。

本书在编排上注重衔接。例如,第 19 章起承上启下的作用,使读者可以很容易地从 C 语言的 struct 过渡到 C++ 的 class。

4. 动手能力

第 16、17 章中介绍了标准 C 函数库的使用,第 30 章介绍了标准模板库的使用,它们都是实际工作中所必须掌握、常用而且重要的知识。

同时,在正文及附录中加强对读者动手能力的培训,强调良好的编码规范,强调自己动手建立项目,并掌握调试技术。

5. 核心概念

C/C++ 中的核心概念包括“传值与传地址”“字符串”“深拷贝与浅拷贝”“动态创建对象”

“安全地使用指针”“面向对象编程思想”等。实际上,如果不深刻理解这些核心概念,就无法写出正确的程序。所以,本书在合适的位置讲解了这些概念,使得读者不必再搜寻别的辅导教材,就可以在学习基本语法的同时掌握这些核心概念。

6. 示例丰富

本书中给出了近三百个示例,为了保证示例的正确性,每个示例都经过作者验证,并且可以在网站(<http://afanihao.cn>,可扫描二维码)上获取完整的示例源码。



7. 丰富的配套资源

1) 在线题库

使用在线题库的方式提供习题,习题按四星级标注,一星为基本练习,三星最难,四星为工业级应用题。每题配备提示、答案或解析,使得读者可以检验自己的思路是否合理。

2) 配套视频

配套 100 集免费视频教程,令学习更加高效和轻松。

3) 源代码下载

近三百个示例源代码免费共享,轻松编写程序。

4) 进阶学习

读者在学习完本书的基本语法之后,可以选择继续学习进阶类教程,使自己具备一个软件开发从业人员的各项基本素质。

5) 答疑与讨论区

以在线的形式提供答疑与讨论区,具体信息请在 <http://afanihao.cn> 上获取。

感谢我的父母、姐姐、哥哥,我的爱人和孩子,感谢他们对我的全力支持。

感谢邵动、邵字红、石广文、李新华、李特梅、姜满如、沈锋平、周建东、郑宇、杨波参与了部分编写工作。

感谢我的老师,张明凤、王善良、周国林、辛燕君、王守玉、申文甲、王羽、王殿兵、郭全、晋宗华、胡振仙、邹明光、顾秉林、朱小梅、李淑琴、张仰森等在我成长之路上对我的栽培。

感谢我的朋友,艾本仁、白建峰、陈刚、陈希、丁将、符汪洋、顾元凯、葛国锋、关正伟、刘星、刘庆生、罗晓宇、裴勇、裴郊区、王浩、邵安馨、石天泽、王静竞、杨浩杰、杨猛、杨禹等给我的帮助。

感谢网友,涂兆昊(神奇的人鱼)、李若滕、伊文斯、张松华(non)、李显发(李小明)、李睿意(lry)、王浩然(连心线)、李显(江南暮春)、卢阿旺(authur)、黄文海(ggev)、神奇海螺、黑夜的忧伤、李泽铭(角逐新梦想)、蔡万成(by)、张强(玄白)、吴霄(stra)、崔祥祥、邓志光等在本教程开发过程中所起的作用。

邵 发

2015 年 8 月于北七家

目 录

第 1 章 开始学习 C/C++	1
1.1 开发平台	1
1.2 第一个程序	1
1.2.1 代码解析	2
1.2.2 在 Windows XP 环境下的问题	2
1.2.3 常见的问题	2
1.3 代码与程序	3
1.4 C 语言和 C++ 语言	3
第 2 章 控制台输入与输出	4
2.1 控制台输出	4
2.1.1 输出整数	5
2.1.2 用变量表示整数	5
2.1.3 指定显示宽度	6
2.1.4 输出小数	7
2.1.5 用变量表示小数	8
2.1.6 指定小数点后的位置	8
2.2 控制台输入	8
2.2.1 输入整数	9
2.2.2 输入小数	9
2.2.3 一次输入多个数	10
2.3 注释	11
2.4 空白	12
2.5 常见问题	12
2.6 综合例题	13
第 3 章 变量与常量	14
3.1 变量	14
3.1.1 变量的定义	14
3.1.2 变量的命名	15
3.1.3 变量的赋值	16
3.2 整型变量	16

3.2.1	char / short / int 类型	16
3.2.2	unsigned 无符号类型	17
3.2.3	* bool 布尔类型	18
3.2.4	* enum 枚举类型	18
3.2.5	* long long 长整数类型	18
3.3	浮点型变量	18
3.4	数的进制表示	19
3.4.1	数的十进制表示	19
3.4.2	数的十六进制表示	19
3.4.3	数的二进制表示	20
3.5	变量与内存	20
3.5.1	二进制存储	20
3.5.2	内存	21
3.5.3	变量的大小	21
3.5.4	变量在内存中的表示	22
3.5.5	变量的地址	22
3.5.6	理解变量的赋值	23
3.6	const 常量	23
3.7	* 字面常量	24
3.8	常用类型的范围	25
第 4 章	数组	26
4.1	引例	26
4.2	数组的定义	26
4.2.1	数组的命名	27
4.2.2	数组的长度	27
4.3	数组的基本使用方法	27
4.3.1	数组的初始值	27
4.3.2	访问数组中的元素	28
4.3.3	用 sizeof 取得数组的大小	29
4.4	数组的内存视图	29
4.5	常见问题	30
4.6	数组的使用实例	31
4.6.1	实例 1	31
4.6.2	实例 2	31
4.6.3	实例 3	31
4.6.4	实例 4	31
4.7	多维数组	32
4.7.1	二维问题的表示	32

4.7.2	二维数组的定义	32
4.7.3	二维数组的初始化	33
4.7.4	二维数组的本质	33
第5章	字符与字符数组	34
5.1	字符是什么	34
5.2	字符的表示	35
5.3	字符的显示	35
5.4	字符常量	35
5.5	字符数组	36
5.5.1	输出字符串	37
5.5.2	输入字符串	37
5.5.3	理解结束符的作用	37
5.5.4	字符串的截断	38
5.5.5	常见问题	38
5.6	转义字符	38
5.6.1	转义字符的概念	38
5.6.2	转义字符的使用举例	39
第6章	表达式与操作符	41
6.1	算术表达式	41
6.2	赋值表达式	42
6.2.1	左值	42
6.2.2	理解赋值运算	42
6.2.3	赋值与算术运算合并	43
6.2.4	等号串连的写法	43
6.3	关系表达式	43
6.4	条件表达式	44
6.5	逻辑表达式	45
6.6	逗号表达式	46
6.7	自增/自减操作符	47
6.8	* 位操作符	48
6.8.1	按位表示	48
6.8.2	位运算规则	48
6.8.3	移位操作	50
6.8.4	应用举例	51
6.8.5	例题: 把 unsigned int 整数转为 4 个字节	52
6.9	类型的转换与提升	52
6.10	优先级与结合顺序	53

6.10.1	使用括号	54
6.10.2	常用的优先级	54
第7章 语句		55
7.1	什么叫语句	55
7.2	if 语句	56
7.2.1	使用复合语句	57
7.2.2	最简形式 if	58
7.2.3	完全形式 if...else if...else if...else	59
7.3	switch 语句	60
7.3.1	匹配	61
7.3.2	跳转与执行	62
7.3.3	注意事项	62
7.4	for 语句	64
7.4.1	引例	64
7.4.2	使用 for 语句	64
7.4.3	变形 1: 省略初始表达式	65
7.4.4	变形 2: 省略第二个表达式	65
7.4.5	变形 3: 省略后置表达式	66
7.4.6	变形 4: 全部置空	66
7.4.7	存在 break 语句	66
7.4.8	存在 continue 语句	66
7.5	while 语句	67
7.5.1	变形: 条件判断内置	68
7.5.2	例题	68
7.6	do...while 语句	69
7.7	综合例题 1	70
7.8	综合例题 2	71
第8章 函数		73
8.1	引例	73
8.2	初步认识函数	73
8.3	函数的定义	75
8.3.1	函数名	75
8.3.2	参数列表	75
8.3.3	返回值类型	76
8.3.4	函数的实现	76
8.3.5	return 语句的用法	76
8.4	函数的调用	77

8.4.1	函数的调用过程	77
8.4.2	参数的顺序	78
8.4.3	函数的传值调用	78
8.4.4	忽略返回值	79
8.4.5	直接使用返回值	80
8.5	全局变量和局部变量	80
8.6	变量的作用域与生命期	81
8.6.1	变量的作用域	81
8.6.2	变量的生命期	82
8.7	变量名重名问题	82
8.8	函数声明与函数定义	84
8.8.1	函数的声明	85
8.8.2	函数声明相关问题	86
8.9	main 函数	86
8.10	参数的隐式转换	86
8.11	* 函数名重载	87
8.12	* 重载函数的匹配	88
8.12.1	精确匹配	88
8.12.2	模糊匹配	88
8.13	* 参数的默认值	90
8.14	* 内联函数	91
8.15	* 函数的递归调用	91
第 9 章	指针	94
9.1	内存地址的表示	94
9.2	指针的概念	94
9.3	星号操作	96
9.4	指针与数组	97
9.4.1	指针加减法	98
9.4.2	指针与数组的转换	98
9.5	指针作为函数的参数	99
9.6	数组作为函数的参数	101
9.7	指针作为函数的返回值	102
9.8	const 指针	103
9.9	void * 型指针	104
9.10	安全地使用指针	105
9.11	* 二重指针	108
9.12	* 二维数组与指针	108
9.12.1	作为函数的参数	109

9.12.2	二维数组与指针的转换	109
第 10 章	结构体	111
10.1	引例	111
10.2	结构体概述	113
10.3	结构体的基本用法	113
10.3.1	变量定义和初始化	113
10.3.2	访问结构体的成员	114
10.3.3	定义结构体数组	114
10.4	结构体的内存视图	115
10.4.1	结构体的成员	115
10.4.2	对齐与填充	116
10.5	结构体的更多用法	117
10.5.1	结构体的赋值	117
10.5.2	结构体指针	117
10.5.3	作为函数的参数	117
10.5.4	作为函数的返回值	118
10.5.5	作为结构体的成员	118
10.6	* 结构体的特殊写法	119
10.6.1	没有类型名	119
10.6.2	函数内部的结构体类型	119
10.6.3	紧凑的形式	120
10.6.4	C 语言中的 struct 定义	120
10.7	结构体的命名	121
10.8	传值与传地址	121
10.9	* 位字段 bit-field	122
10.10	实例	123
第 11 章	* 联合体	125
11.1	概念	125
11.2	进一步理解 union	126
11.3	实例	127
11.4	union 的替代方案	128
第 12 章	动态分配内存	130
12.1	引例	130
12.2	动态分配内存	130
12.3	内存管理器与堆	131
12.4	对象的生命期	133

12.5	实例 1	134
12.6	实例 2	135
12.7	常见问题	136
第 13 章	链表	138
13.1	概念	138
13.2	链表的构造	138
13.2.1	头节点与末节点	139
13.2.2	链表头的作用	139
13.2.3	常见问题	139
13.3	有头链表	140
13.3.1	概念	140
13.3.2	定义一个有头链表	140
13.3.3	添加一个节点	141
13.3.4	有头链表的遍历	142
13.3.5	按顺序插入节点	142
13.3.6	查找和删除节点	143
13.3.7	和无头链表的比较	143
第 14 章	引用	144
14.1	引用的定义	144
14.2	与指针的区别	144
14.3	简单的例子	145
14.4	作为函数的参数	146
14.5	作为函数的返回值	147
14.6	const 引用	148
第 15 章	字符串	149
15.1	字符串的三种形式	149
15.2	字符串常量的多行表示	150
15.3	字符串与普通数据	151
15.4	字符串的遍历	151
15.5	字符串长度	152
15.6	字符串复制	153
15.7	字符串比较	154
15.8	字符串插入和删除	155
15.8.1	删除字符	155
15.8.2	插入字符	156
15.8.3	插入和删除的效率问题	156

15.9	字符串的分割	157
15.10	用数组还是用指针	159
第 16 章 标准 C 函数库		162
16.1	stdio.h	162
16.2	math.h	164
16.3	time.h	165
16.3.1	time_t	166
16.3.2	struct tm	166
16.3.3	time 取得系统当前时间	167
16.3.4	localtime 得到年月日时分秒	167
16.3.5	mktime 构造时间	168
16.4	stdlib.h	169
16.5	string.h	171
第 17 章 文件操作		175
17.1	认识文件	175
17.2	保存数据	176
17.2.1	fopen 打开文件	176
17.2.2	fclose 关闭文件	177
17.2.3	fwrite 写入数据	177
17.3	读取数据	177
17.4	数据的存储格式	179
17.5	存储格式：按字节存储	180
17.5.1	指针指向的对象	181
17.5.2	Run-Length Encoding 存储	182
17.6	存储格式：文本化存储	183
17.6.1	fprintf 按行格式化写入	183
17.6.2	fgets 按行读取	184
17.7	文件的随机访问	184
17.7.1	fseek 随机访问	184
17.7.2	文件位置指示器	185
17.7.3	随机访问示例	185
17.7.4	fseek 的物理限制	186
17.7.5	文件被重复打开的情况	186
17.8	* 文件打开模式	186
17.9	* 常见问题	187
第 18 章 多文件项目及编译过程		188
18.1	extern	188

18.1.1	extern 声明全局函数	188
18.1.2	extern 声明全局变量	189
18.1.3	深入理解 extern	190
18.2	多文件项目的生成	190
18.2.1	第一阶段: 编译	190
18.2.2	第二阶段: 链接	191
18.2.3	* 用伪代码表示整个过程	192
18.2.4	全量编译与增量编译	193
18.3	头文件 #include 指令	193
18.3.1	为什么需要头文件	193
18.3.2	使用头文件	195
18.3.3	#include 指令的原理	196
18.3.4	头文件的重复包含问题	196
18.3.5	头文件里的内容	197
18.4	宏定义 #define 指令	197
18.4.1	#define 一个数值	198
18.4.2	#define 一个算式	199
18.4.3	几个常见的宏定义	200
18.5	条件编译指令 #if	200
18.5.1	#if... #endif	200
18.5.2	#ifdef... #endif	200
18.5.3	解决头文件重复包含的问题	201
18.6	* main 函数的参数和返回值	202
18.6.1	main 函数的参数	202
18.6.2	main 函数的返回值	204
18.7	* static 的用法	204
18.7.1	static 修饰变量	204
18.7.2	static 修饰函数	205
第 19 章	面向对象编程	206
19.1	面向对象设计的过程	206
19.2	实例演示	206
19.2.1	需求与设计	206
19.2.2	写出代码框架	207
19.2.3	实现各个函数接口	209
19.3	封装	210
第 20 章	类	212
20.1	类和成员变量	212

20.1.1	了解访问修饰符	213
20.1.2	访问修饰符的用法	213
20.2	类和成员函数	214
20.2.1	访问修饰符的限制	215
20.2.2	了解 this 指针	216
20.2.3	this 指针的用法	217
20.3	变量名字的覆盖	218
20.4	命名规范	220
20.4.1	类名的命名规范	220
20.4.2	成员函数的命名规范	221
20.4.3	成员变量的命名规范	221
20.5	类的封装	222
20.6	类的分离式写法	223
20.7	* const 对象与 const 函数	224
20.7.1	const 成员函数	224
20.7.2	const 版本与非 const 版本	224
第 21 章 构造与析构		226
21.1	引例	226
21.2	构造函数	226
21.2.1	重载构造函数	227
21.2.2	构造函数的调用	227
21.3	析构函数	229
21.3.1	析构函数的调用	229
21.3.2	在析构函数中释放资源	230
21.4	自动生成的构造/析构函数	231
21.5	默认构造函数	232
21.6	构造函数的初始化列表	232
21.6.1	参数名与成员变量名可以相同	233
21.6.2	成员本身也是 class 类型	233
21.6.3	混合使用两种初始化方式	234
21.6.4	必须使用初始化列表的情形	234
21.7	构造与析构的顺序	235
21.8	分离式写法	236
21.9	* 无名对象	237
21.10	* 构造函数与类型转换	237
第 22 章 动态创建对象		239
22.1	回顾 malloc/free	239

22.2	用 new/delete 创建/销毁对象	240
22.3	new/delete 与 malloc/free 的区别	241
22.4	为 new 指定初始化参数	241
22.5	默认构造函数的必要性	243
22.6	注意事项	243
第 23 章	继承	244
23.1	引例	244
23.2	继承的概念	244
23.3	访问修饰符 protected	246
23.4	成员函数的重写	247
23.5	虚拟继承	248
23.5.1	父类指针指向子类对象	248
23.5.2	虚函数 virtual	249
23.5.3	virtual 析构函数	250
23.6	继承关系下的构造与析构	250
23.7	* 多重继承	251
23.8	* 继承函数与纯虚类	252
23.9	* 以 protected/private 方式继承	253
第 24 章	拷贝构造函数	254
24.1	定义	254
24.2	拷贝构造函数的调用	254
24.3	默认的拷贝构造函数	256
24.4	定义拷贝构造函数	256
24.5	深度拷贝	258
第 25 章	静态成员	260
25.1	static 定义全局变量	260
25.2	static 定义全局函数	261
25.3	与普通成员的区别	263
25.4	static 语法的特点	264
25.5	应用 1: 实例计数	265
25.6	应用 2: 单例模式	267
第 26 章	朋友成员	269
26.1	引例	269
26.2	类的朋友	269
26.3	friend 的语法	270

26.4	实例	270
26.5	进一步讨论	271
第 27 章 重载操作符		273
27.1	引例	273
27.2	算术操作符	274
27.2.1	参数类型	275
27.2.2	加法的互换性	275
27.3	赋值操作符 =	276
27.3.1	赋以不同的类型	276
27.3.2	深拷贝	277
27.3.3	相关问题及注意事项	278
27.3.4	检查项列表	279
27.4	自增操作符 ++ 与自减操作符 --	279
27.5	关系操作符	280
27.6	逻辑操作符	281
27.7	类型转换操作符()	281
27.8	元素操作符[]	282
27.9	输入输出操作符 >> 与 <<	283
27.10	操作符 new 与 delete	284
27.11	小结	284
第 28 章 内部类和名字空间		285
28.1	内部类	285
28.1.1	内部类的语法	285
28.1.2	内部类和普通类的区别	286
28.1.3	内部类的用途	286
28.1.4	内部类的分离式写法	286
28.1.5	内部 enum 类型	287
28.2	名字空间	288
28.2.1	名字空间的定义	288
28.2.2	分离式写法	289
28.2.3	using 语句: 解除前缀	290
28.2.4	名字空间里的其他类型	291
28.2.5	名字空间的应用场景	291
第 29 章 模板		292
29.1	函数模板	292
29.2	类模板	294