

# C/C++

## 程序设计教程

郭小刚 金星 编著

 人民邮电出版社  
POSTS & TELECOM PRESS



C/C++

# 程序设计教程

郭小刚 金星 编著

人民邮电出版社



## 图书在版编目 (CIP) 数据

C/C++程序设计教程/郭小刚, 金星编著. —北京: 人民邮电出版社, 2004.1

ISBN 7-115-11931-7

I. C... II. ①郭... ②金... III. C语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 099424 号

## 内 容 提 要

本书结合最新的 C/C++ 标准, 对 C/C++ 程序设计语言进行深入浅出的介绍。从混合程序设计的角度理顺 C 和 C++ 程序设计语言间的异同。从最基本的概念出发, 介绍 C/C++ 语言的来龙去脉, 并且一步步地进入语言的更深层次开发。全书精心对比相似语法的不同特点, 列举了大量实例, 深刻剖析隐晦难懂之处, 力求使读者从根本上掌握 C/C++ 语言。

本书具备教材的条理清晰、逻辑严谨, 同时力求手册的全面系统。本书前部分将 C++ 作为一个更好的 C 来描述。通过结构化程序设计的学习, 读者可具备软件开发所需要的基本知识。针对面向对象理论的编程方法, 本书对封装、继承和多态、运算符重载和模板语法现象等提供了明确而细致的解说。

本书力图做到宽口径、厚基础、高起点, 适用从入门到精通的各个层次的 C/C++ 语言学习者和软件开发者的实际需要, 减少入门者不必要的摸索时间。本书可作为高等院校理工科各专业 C 或 C++ 程序设计语言的教材, 也可供自学者学习 C 或 C++ 语言使用。

## C/C++程序设计教程

- ◆ 编 著 郭小刚 金星  
责任编辑 王文娟
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线 010-67132692  
北京汉魂图文设计有限公司制作  
北京顺义振华印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16  
印张: 25  
字数: 602 千字 2004 年 1 月第 1 版  
印数: 5 001—7 000 册 2004 年 8 月北京第 2 次印刷

ISBN 7-115-11931-7/TP · 3758

定价: 32.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223



# 前 言

## (一) 本书的目的

C++语言本身的目标是将优秀的C语言和面向对象理论整合在一起,提供给开发软件的程序员使用。C语言和C++语言的关系是如此紧密:C是C++的子集,C++是C的超集,C++是一个更好用的C。C语言作为操作系统事实上的开发语言,是C++语言厚实的基础。因此本书视C语言与C++语言为一种动态发展的语言而非两种本质不同的语言,把它们结合在一起进行阐述。

渊源于C/C++演化出两种重要语言——Java和C#。学会了C/C++语言之后学习Java和C#会更加轻车熟路,而C/C++语言比Java具有更纵深的处理系统资源的能力。

C++语言最得意的手笔就是将功能强大、运算快捷与高效灵活的C语言作为其子集,C语言开发的C运行库函数可以不加改动地在C++语言环境中使用,达到了软件省时省力的重用。

C++继承或借鉴了FORTRAN高效的变量引用调用约定,又作为C语言的超集,本身是一个和谐的混合编程环境。例如MFC类库成员函数中大量地包含了C运行库函数的调用,同时在关键的消息映射函数中还内嵌汇编语言的运用,因此MFC类库作为面向对象编程事实上的范例标准,本身就是典型的混合编程的产物。

鉴于C++语言代表着以往语言的精华和提炼,而目前市面上的C++教材有不少片面强调面向对象理论的优点,从而导致观点的偏狭。不少C++入门书籍在基本控制流程、基本表达式的阐述以及联合概念的阐述上过于简略,而这些运算和概念是基础性的。本书作为一本独立的入门教材,在基本概念的阐述上尽量细致入微。

单一的C编译器已经淡出市场,而目前一些C教材依然沿用的是15年前陈旧的C语法,不能适应在新的C++编译器中进行软件开发的需要。从C++语法视点看旧的C教材,存在明显的类型归属含糊的问题。C++语言矫正了早期C语言的不少潜在的类型匹配的错误,而这在目前C教材中并未得到应有的反映。因此本教材力求反映ANSI C(C89,C99)/C++(98)的最新标准,不是用纯粹的面向对象的观点而是从混合语言的角度返朴归真地阐述C++语言的本来面目。本书的目的在于C/C++语言教学的平稳承接。目前流行的C教材难以适应新的混合编程的编译环境。

从程序员的角度看,C/C++程序设计中最重要的是以下三个面向:一是面向系统提供的API函数或库函数,二是面向具体的专业问题,三是面向软件的用户。

针对自学者的特点兼顾软件开发的实际需要,本书力求深入透彻地阐述语法现象,特别是着重说明相似语法与术语的相同与差异,使读者在学习本书之后能迅速地转入实际的各个层次的软件开发。

本书定位于程序设计的初学者,对于有经验的教师、科技工作者本书也具有参考价值。本书适合作为大学本科(或大专)计算机专业和其他理工科专业C/C++程序设计教材。通过本书的学习可以达到如下的目标:

1. 重点掌握C/C++语言的语法,懂得混合程序设计的方法。



2. 能够灵活地运用 C/C++ 语言提供的语法手段解决具体的工程问题。
3. 从多视点全方位的角度把握面向过程和面向对象编程的特点。

本书的价值在于全面阐述 C 语言的优点，在 C++ 语言的视野上予以提高，做到宽口径、厚基础、高起点，适用从入门到精通的各个层次的 C/C++ 语言学习者和软件开发者的实际需要，减少入门者不必要的摸索时间。

## (二) 本书的特色

本书由浅入深地讲述 C/C++ 语言的语法，重点概念得到强化性的介绍，尽量做到术语专用而不混用术语。本书的特点和新颖之处如下。

1. 函数。函数是 C/C++ 程序设计最重要的基础。本书在第 1 章便引进函数的概念，随后的章节逐步加强函数调用的匹配和算例阐述，第 8 章对函数进行深入系统地介绍。

2. 指针和数组。指针和数组是密不可分的。本书对指针和数组的相互关系进行深入而清晰的对比和归类，穿插地介绍指针和数组的相似和差异之处，对指针的属性、指针与变量生存期的相互关系进行了详细的介绍。例如只读指针被含糊地翻译为指向常量的指针，而只读指针实际上可以指向变量或左值数组占有的数据区。并且对函数指针（包括指向全局函数的指针和指向成员函数的指针）进行了全面的介绍。

3. 引用。引用是 C++ 新引进的，引用是变量的别名。本书精心对比指针、引用和变量，介绍“引用”在 C++ 语言中的作用。“引用”一词在本书专指变量的别名，而不混用“引用”概念。索引在编译器的角度是找到某一标识符并使用它，动词引用的概念是使用标识符，本书用索引代替普通的引用概念。

4. 混合程序设计。C++ 语言本身是一门混合程序设计的语言。本书强调混合程序设计而非纯粹的面向对象编程。因此本书的常用算法用 C 语言面向过程的方式提供。面向对象部分重点在于阐述语法现象。“联合”作为一种重要的内存动态覆盖技术与课题组软件开发合作技术在本书得到详细的阐述。通过介绍 C++ 代码幕后还原为 C 代码的机制，读者容易在面向过程编程和面向对象编程两者之间进行转换，进而达到混合编程两者合一的自由境界。

5. 文件处理是 C 语言教学的重点，C++ 的输入输出类库是 C++ 语言教学的重点，两者是平行的，学其一即可。通常 C 教材介绍文件处理，C++ 教材介绍输入输出类库，本书两者都提供，以不失内容的完整，并提供自学的便利。

6. const 关键字修饰的变量称为不变量或常量。本书进一步根据 const 初始化源数据的不同，将其严格地分为静态常量和活动的不变量。对于前置运算符 ++L 和后置运算符 L++ 的细微异同进行了介绍。对于其他运算符也进行全面的系统的阐述。关于算术负运算和补码表示之间的关系进行了清晰的刻画。

7. 数据信息的隐蔽本质上是编译器的责任，在确保数据安全到位的前提下，本书侧重信息的共享和流通。本书介绍了全局指针或返回指针的函数如何将静态变量或静态函数跨模块共享，介绍了指针如何在外部访问私有数据，如何改写只读数据区等。只有知道如何突破 C/C++ 语言的信息屏蔽机制，知道屏蔽机制作用的有限性，才可以在其框架内恰当地实施数据的独立和互不干涉。

8. 多态在国外的权威书籍中用于指基对象的指针调用派生类的虚函数。本书遵循国际的



通用分类，而不是将函数重载、运算符重载和隐含类型转换等也归于多态。尽量做到一个术语专用于一个概念。

9. 为了读者能够迅速掌握 C/C++ 程序设计的实用方法，学以致用，本书提供了丰富的算例。全部算例在 Visual C++ 6.0 上编译通过。本书重点倾向于体现软件工程的基本原则，强调编写高效清晰的程序。将程序的高效、可移植性和简单明了以及解决问题摆在软件开发的首要地位。

### (三) 本书的编排和使用

全书的内容以最新的 ANSI C/C++ 标准为参照，同时紧密结合 C++ 语言源于 C 语言的事实，综合介绍 C 和 C++ 程序设计中涉及到的语法现象。从软件开发的实际需要出发，内容几乎涵盖了 C 语言和 C++ 语言的所有重要部分（没有包括汇编接口技术）。经过精心设计和挑选的算例尽量体现成熟的经典算法。

本书分两部分，面向过程的部分和面向对象的部分。本书前 13 章包括 C 语言的基本语法和 C++ 对于 C 语言的改进即面向过程的部分，前 13 章覆盖了目前 C 语言教材的内容，因此前 13 章可以满足改良的 C 语言的教学需要。后面的 11 章是面向对象的内容。C 语言的基本内容要求讲课学时至少为 48 学时，面向对象的部分要求 24~32 个学时。本书丰富的算例可以满足教学和软件开发的各种需要，可以满足学时少的专业的学生课外自学的需要。

学过 C 语言的读者也可用本书作为学习 C++ 语言的教材。标有“C++ 特有”内容的章节，如第 2 章中的引用概念，第 4 章中的 C++ 输入输出流，第 8 章、第 9 章、第 14~24 章的内容可以满足 C++ 面向对象的学习需要。两种语言的交集部分可以在温故知新的阅读中得以提高。例如第 6 章“指针与数组”、第 7 章中的返回指针值的函数与生存期，这些 C 语言的重要内容，本书在新的高度进行了全面系统的解说。

标有“知识拓展”字样的章节可以作为深入学习的素材。跟以“查阅”字样的章节表示 C++ 新增的内容或不是高频采用的内容，它们可供编程参考。后面跟以“多学时”字样的章节对于编程而言是高频采用的语法和算法，建议根据学时选学或自学。灵活是 C 语言的精神也是 C++ 语言的精神，老师可以利用本书围绕精选的算例灵活地讲解 C/C++ 程序设计，学生可以灵活地自学。

本书在确保 C/C++ 代码可读性的前提下，采取了节省纸张的编排方式。书中黑体标注的内容是特别重要的。楷体标注的内容表示语言的变迁、语法的细节规定、相似概念的差异比较、编译器的具体实现以及语法概念的提前介绍等。楷体标注的有些内容在初学时可不必立即深究，可作为编程的参考或视野的开拓。

本书涉及到许多深入的细节，读者可在软件开发、上机实验中逐步理解掌握。初学者开始的时候不必拘泥于语法细节，应该首先在总体上把握语言基本的内容，然后逐步深入进而融会贯通。

为方便老师教学和读者自学，我们为本书制作了相应的电子教案，同时还提供了本书的全部算例和练习题答案，这些资料可到人民邮电出版社网站下载，网址为 [www.ptpress.com.cn/download](http://www.ptpress.com.cn/download)，点击进入“计算机类”即可。

本书第 23 章和第 24 章由金星著写，其余由郭小刚执笔，最后由郭小刚统稿。在本书编著过程中，张平博导、张俊彦博士给予的深切关心、始终如一的鼓励，以及尖锐独到的建议







# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 机器语言、汇编语言和高级语言 .....	1
1.2 C 语言的历史与特点 .....	2
1.3 从 C 语言到 C++ 语言的进化 .....	3
1.4 计算机结构的五个主要单元 .....	3
1.5 C/C++ 程序的实现 .....	4
1.6 C/C++ 源程序的基本形式 .....	5
1.7 函数初步 .....	8
1.8 C/C++ 程序上机步骤简介 .....	11
1.9 小结 .....	12
<b>第 2 章 基本元素、类型和概念</b> .....	14
2.1 基本语言元素 .....	14
2.1.1 C/C++ 中的字符集 .....	14
2.1.2 语言符号 .....	14
2.2 基本数据类型 .....	16
2.3 算术负运算与补码转换 .....	18
2.4 常数和数据的内存布局 .....	21
2.5 sizeof 运算符 .....	26
2.6 变量 .....	27
2.7 格式化输出函数 printf 和输入函数 scanf .....	29
2.7.1 输出函数 printf .....	29
2.7.2 输入格式转换函数 scanf .....	33
2.8 const 量或不变量 .....	35
2.9 变量的引用 (C++ 特有) .....	36
2.10 typedef 类型声明语句 .....	37
<b>第 3 章 运算符与表达式</b> .....	39
3.1 运算符与表达式概述 .....	39
3.2 左值和右值 .....	40
3.3 运算符的优先级与结合性 .....	40
3.4 算术运算表达式 .....	41
3.5 混合运算时数据的类型转换 .....	42



3.5.1	强制类型转换 .....	42
3.5.2	常用的算术转换 .....	43
3.5.3	隐含类型转换 .....	44
3.6	赋值表达式和混合赋值 .....	45
3.7	算术复合赋值表达式 .....	47
3.8	自增与自减运算符表达式 .....	49
3.9	逗号运算符表达式 .....	50
3.10	表达式的副作用 .....	51
3.11	位操作运算符 (多学时) .....	52
<b>第 4 章</b>	<b>逻辑运算和选择控制语句 .....</b>	<b>56</b>
4.1	语句概述 .....	56
4.2	C++中特有的输入输出流 .....	58
4.3	逻辑判断 .....	60
4.3.1	关系运算符 .....	60
4.3.2	逻辑运算符 .....	61
4.4	选择语句 .....	63
4.4.1	单路分支 if 语句 .....	63
4.4.2	双路分支 if-else 语句 .....	65
4.4.3	if-else 语句或 if 语句的嵌套 .....	66
4.4.4	if-else if-else 语句 .....	67
4.4.5	条件运算符 .....	69
4.5	switch 语句 .....	70
<b>第 5 章</b>	<b>循环控制 .....</b>	<b>73</b>
5.1	while 语句 .....	73
5.2	do-while 语句 .....	74
5.3	for 语句 .....	76
5.4	break 语句与 continue 语句 .....	78
5.5	goto 语句 .....	80
5.6	多重循环 .....	81
<b>第 6 章</b>	<b>指针与数组 .....</b>	<b>83</b>
6.1	指针的概念 .....	83
6.2	一级指针和指针的操作 .....	84
6.3	指针与间接变量 .....	85
6.4	一维数组 .....	86
6.4.1	一维数组的定义 .....	87
6.4.2	一维数组的初始化 .....	87



6.4.3	一维数组的内存映像和指针加减关系运算	88
6.4.4	函数的数组形参和指针形参	91
6.5	const 指针	93
6.6	二维数组	95
6.6.1	二维数组的定义	95
6.6.2	二维数组的初始化	97
6.6.3	多维数组	97
6.7	指向数组的指针	98
6.8	二级指针	101
6.9	指针数组	102
6.10	void 关键字与 void *型的指针	105
6.11	程序动态存储结构	107
6.11.1	C++中 new 运算符和 delete 运算符	107
6.11.2	C 语言中的动态内存分配函数 malloc 和 free 函数	108
6.12	指针的类型转换和匹配关系 (知识拓展)	111
6.13	下标表达式与访问指针寻址计算 (知识拓展)	113
<b>第 7 章 程序的结构</b>		<b>116</b>
7.1	程序设计方法	116
7.1.1	自顶向下的程序设计	116
7.1.2	自底向上的程序设计	118
7.1.3	结构化程序设计	118
7.1.4	多文件结构	119
7.2	作用域或范围 scope	119
7.3	可见性与名称隐藏	122
7.4	生存期与存储属性	123
7.5	extern 关键字与外部连接属性	124
7.6	static 关键字与内部连接属性	125
7.6.1	静态函数或内部函数	125
7.6.2	静态全局变量和静态局部变量	126
7.7	返回指针值的函数与生存期	127
7.8	枚举	130
<b>第 8 章 函数</b>		<b>134</b>
8.1	函数与调用约定	134
8.2	函数的总体概念	135
8.2.1	函数的返回类型	136
8.2.2	函数的定义	136
8.2.3	函数原型	137



88	8.2.4 return 语句	138
10	8.2.5 函数的使用	139
80	8.3 内联函数 (inline function)	140
20	8.4 引用和数值传递方式具体比较	141
20	8.5 函数重载——function overloading (C++特有)	143
70	8.5.1 重载函数与名称细分	143
70	8.5.2 函数虚实结合类型匹配 (知识拓展)	145
80	8.5.3 连接 C 语言中的程序模块 (查阅)	146
101	8.6 默认参数——default argument values (C++特有)	147
501	8.7 函数的调用机制 (多学时)	148
201	8.8 函数的嵌套与递归	150
301	8.9 数组的排序和查找 (根据学时酌情选讲)	152
301	8.9.1 直接插入排序	152
801	8.9.2 选择排序	153
111	8.9.3 交换法实现冒泡排序	154
811	8.9.4 二分查找	155
	<b>第 9 章 函数指针</b>	<b>157</b>
011	9.1 函数指针的定义	157
011	9.2 函数指针的使用	158
811	9.3 函数指针作为形参	159
811	9.4 typedef 关键字、函数指针与重载函数 (多学时)	163
011	9.5 函数指针构成的数组 (多学时)	164
	<b>第 10 章 预处理过程</b>	<b>168</b>
531	10.1 编译预处理指令概述	168
151	10.2 预处理指令#include	169
251	10.3 #define 指令	169
251	10.3.1 #undef 取消标识符定义	169
051	10.3.2 不带参的宏替换	169
751	10.3.3 带参的宏定义	172
001	10.4 宏调用和内联函数 (知识拓展)	173
151	10.5 条件编译指令 (多学时)	174
151	10.6 字符串预处理操作符 (多学时)	177
	<b>第 11 章 字符和内存处理</b>	<b>180</b>
081	11.1 字符数组、指针和字符串的初始化作用	180
081	11.2 strlen 函数确定字符串有效长度	185
381	11.3 strcpy 函数拷贝字符串	186



11.4	strcat 函数合并字符串	187
11.5	strcmp 函数比较字符串	187
11.6	字符串排序算例	188
11.7	memcpy 函数拷贝内存	189
11.8	程序的入口函数 main	190
<b>第 12 章 结构与联合</b>		<b>193</b>
12.1	结构的声明和结构变量	193
12.2	结构变量的定义形式	194
12.3	对结构数据的操作	195
12.4	结构变量的内存分布	197
12.5	结构的初始化	198
12.6	结构与函数	200
12.7	结构的组合	202
12.8	单链表 (多学时)	204
12.8.1	单链表概述	204
12.8.2	链表结构的建立	204
12.8.3	动态建立堆中链表	207
12.8.4	对链表的插入	208
12.8.5	链表结点的脱离	211
12.9	联合 union	213
12.9.1	联合的特性和定义	213
12.9.2	联合的内存映像	215
12.9.3	无名联合 (知识拓展)	217
12.10	数据的引用类型转换 (知识拓展)	219
12.11	位域或位字段 (查阅)	221
<b>第 13 章 文件</b>		<b>223</b>
13.1	文本流和二进制流	223
13.2	流文件	224
13.3	文件的打开函数 fopen 和关闭函数 fclose	225
13.4	格式读写 fprintf 和 fscanf 函数	227
13.5	出错的测试或清除函数 (feof, ferror, clearerr)	229
13.6	字符和字符串读写函数	229
13.6.1	读取单个字符的 fgetc 函数	229
13.6.2	存写单个字符的 fputc 函数	230
13.6.3	按行读文本串 fgets 函数	230
13.6.4	按行写文本串 fputs 函数	231
13.7	无格式转换的读写函数 fread 和 fwrite	231



13.8	文件的定位 .....	233
13.8.1	ftell 函数告知当前位置 .....	234
13.8.2	fseek 函数探寻文件的位置 .....	234
13.8.3	rewind 函数反绕到文件开头位置 .....	234
13.9	一个简单的读写存盘程序 .....	235
<b>第 14 章</b>	<b>迈入面向对象编程部分 .....</b>	<b>239</b>
14.1	面向对象理论鸟瞰 .....	239
14.2	面向对象程序设计的概念 .....	240
14.3	面向过程和面向对象编程 .....	241
14.4	类的声明 .....	243
14.5	对象的定义 .....	244
14.6	成员函数与 this 关键字 .....	245
14.7	类作用域和成员的访问 .....	246
14.8	内联成员函数 .....	249
14.9	前置说明 (forward reference) .....	249
14.10	函数的接口转换 (知识拓展) .....	250
<b>第 15 章</b>	<b>特殊的成员函数 .....</b>	<b>253</b>
15.1	构造函数 (constructor) .....	253
15.1.1	构造函数的特殊性 .....	253
15.1.2	拷贝构造函数和无参构造函数 .....	254
15.1.3	缺省构造函数 .....	255
15.2	构造函数的调用 .....	255
15.2.1	对象名调用 .....	256
15.2.2	new 运算符调用 .....	257
15.2.3	无名对象调用 .....	257
15.2.4	函数对象名语法算例 .....	257
15.3	对象数组的初始化 .....	258
15.4	析构函数 (destructor) .....	259
15.5	赋值运算符函数 operator= .....	261
15.6	成员函数重载 .....	262
15.7	缺省参量的成员函数 .....	263
15.8	编译器默默提供的成员函数 (知识拓展) .....	264
<b>第 16 章</b>	<b>数据的共享和流通 .....</b>	<b>266</b>
16.1	浅拷贝和深拷贝 (shallow copy 和 deep copy) .....	266
16.2	只读成员函数与 volatile、mutable 关键字 .....	268
16.3	友元 (friend) .....	270



16.3.1	友元函数	271
16.3.2	友元类	272
16.4	静态成员	273
16.4.1	静态数据成员	273
16.4.2	静态成员函数	274
16.5	指向成员的指针 (多学时)	276
16.5.1	指向数据成员的指针	276
16.5.2	指向成员函数的指针	277
<b>第 17 章</b>	<b>运算符重载</b>	<b>280</b>
17.1	运算符重载的概念	280
17.2	禁止重载的运算符	281
17.3	运算符重载的规则	282
17.4	单目运算符函数	282
17.5	双目运算符函数	284
17.6	数据的类型转换	285
17.6.1	单参数构造函数的类型转换	286
17.6.2	explicit 关键字抑制自动类型转换	286
17.6.3	运算符类型转换函数	287
17.7	自增运算符和自减运算符	288
17.8	函数调用运算符()函数 (多学时)	290
17.9	访问成员运算符-> (多学时)	291
17.10	数组下标运算符[] (多学时)	292
17.11	枚举类变量的运算性质 (查阅)	294
<b>第 18 章</b>	<b>组合和继承</b>	<b>296</b>
18.1	组合	296
18.1.1	组合的概念	296
18.1.2	引用型成员和 const 成员	296
18.1.3	嵌入对象的初始化	297
18.1.4	冒号初始化语法	298
18.2	继承和派生	299
18.2.1	继承的概念	299
18.2.2	间接基类和间接派生类	300
18.3	派生类的声明和对象定义	300
18.4	派生类的三种继承方式	301
18.5	继承和不继承的语义	302
18.6	派生与继承的算例	303
18.7	构造和析构的次序	304



<b>第 19 章 多态和虚函数</b> .....	307
19.1 函数覆盖和函数重载 .....	307
19.2 虚函数的声明 .....	308
19.3 多态类层次之间的适应关系 .....	308
19.3.1 对象赋值兼容规则 .....	308
19.3.2 静态类型和动态类型 .....	309
19.4 静态联编和动态绑定 .....	309
19.5 名称索引的优先级 .....	311
19.6 虚函数动态绑定调用算例 .....	312
19.7 虚拟析构函数 .....	314
19.8 纯虚函数和抽象类 .....	315
<b>第 20 章 模板</b> .....	318
20.1 模板的概念和方法 .....	318
20.2 函数模板 .....	319
20.2.1 函数模板的机制 .....	319
20.2.2 函数模板的声明 .....	319
20.2.3 函数模板的实现 .....	319
20.3 函数模板和函数重载算例 .....	320
20.4 非模板的重载函数 .....	321
20.5 类模板 .....	322
20.5.1 类模板的声明 .....	322
20.5.2 类模板成员函数的描述 .....	322
20.5.3 类模板的具体实现 .....	323
20.5.4 类模板的成员函数显式实现 .....	324
20.6 类模板的默认参数 .....	324
<b>第 21 章 异常处理技术</b> .....	326
21.1 异常处理的概况 .....	326
21.2 C++异常处理的途径 .....	327
21.3 异常的多路捕获 .....	330
21.4 异常的重新抛出 .....	331
21.5 对象的清理 .....	332
21.6 类层次的异常处理策略 .....	334
21.7 关于抛出异常的函数显式说明 (查阅) .....	337
<b>第 22 章 多重继承和类型变换</b> .....	338
22.1 多个直接基类 .....	338



22.2	虚拟基类	340
22.3	多继承的构造函数	341
22.4	名称的二义性	344
22.5	新的类型变换 (查阅)	345
22.5.1	dynamic_cast 动态类型变换	346
22.5.2	static_cast 静态类型变换	348
22.5.3	reinterpret_cast 重新翻译变换	350
22.5.4	const_cast 松动 const 属性变换	350
22.5.5	typeid 关键字	351
<b>第 23 章</b>	<b>C++ 的输入输出类库</b>	<b>355</b>
23.1	iostream 类层次概述	355
23.2	关于 C++ 的新类	356
23.3	流对象的输入输出重载函数	356
23.4	重载流插入运算符和流提取运算符	358
23.5	格式化的输入输出	359
23.5.1	操作算子 (manipulator) 的概念	359
23.5.2	整数转换的操作算子 hex, oct, dec	359
23.5.3	自定义操作算子	360
23.5.4	格式化标志值	360
23.5.5	控制浮点精度	361
23.5.6	浮点数和科学计数法	362
23.5.7	字符填充和宽度、对齐控制	363
23.6	流对象磁盘文件的输入输出	364
23.6.1	文件的打开函数	364
23.6.2	文件的关闭函数 close	365
23.6.3	测试状态的函数 (bad, eof, good, clear 等)	366
23.6.4	字符和字符串读写函数	367
23.7	无格式转换的读写函数 read 和 write	368
23.7.1	read 读取函数	368
23.7.2	write 存写函数	369
23.8	文件的定位	370
<b>第 24 章</b>	<b>名称空间和嵌套类 (查阅)</b>	<b>372</b>
24.1	名称空间 (namespace)	372
24.1.1	固有的名称空间	372
24.1.2	namespace 构建名称空间	373
24.1.3	using 声明语句	374
24.1.4	名称空间的分散布置	375



24.1.5	名称空间的歧义性	376
24.1.6	嵌套的名称空间	377
24.1.7	名称空间的别名	377
24.1.8	无名的名称空间	378
24.2	嵌套类	379
24.3	友元函数和嵌套类	381
附录	ASCII 码表	383
23.1	iostream 类层次概述	1.2.1
23.2	关于 C++ 的流类	1.2.2
23.3	流对象的输入输出重载函数	1.2.3
23.4	重载流插入运算符和流提取运算符	1.2.4
23.5	格式化的输入输出	1.2.5
23.5.1	操作符 (manipulator) 的概念	1.2.5.1
23.5.2	整数转换的操作符 hex, oct, dec	1.2.5.2
23.5.3	自定义操作符	1.2.5.3
23.5.4	格式化的标志	1.2.5.4
23.5.5	控制浮点精度	1.2.5.5
23.5.6	浮点数和科学计数法	1.2.5.6
23.5.7	字符填充和宽度、对齐控制	1.2.5.7
23.6	流对象磁盘文件的输入输出	1.2.6
23.6.1	文件的打开函数	1.2.6.1
23.6.2	文件的关闭函数 close	1.2.6.2
23.6.3	测试状态的函数 (bad, eof, good, clear 等)	1.2.6.3
23.6.4	字符和字符串读写函数	1.2.6.4
23.7	无格式转换的读写函数 read 和 write	1.2.7
23.7.1	read 读取函数	1.2.7.1
23.7.2	write 写入函数	1.2.7.2
23.8	文件的定位	1.2.8
24.1	名称空间 (namespace)	1.3.1
24.1.1	固有的名称空间	1.3.1.1
24.1.2	namespace 构建名称空间	1.3.1.2
24.1.3	using 声明	1.3.1.3
24.1.4	名称空间的成员变量	1.3.1.4