



万水计算机技术实用大全系列



[美] Kris Jamsa 著

张春晖 刘大庆 李 越 等译

C/C++/C# 程序员实用大全

— C/C++/C# 最佳编程指南 (精华版)

Jamsa's C/C++/C# Programmer's Bible
The Ultimate Guide to C/C++/C# Programming

源代码可从中国水利水电出版社网站下载

ONWORD PRESS

THOMSON LEARNING



中国水利水电出版社
www.waterpub.com.cn

万水计算机技术实用大全系列

C/C++/C#程序员实用大全

——C/C++/C#最佳编程指南

(精华版)

[美] Kris Jamsa 著

张春晖 刘大庆 李 越 等译

利水文化 审校

中国水利水电出版社

内 容 提 要

本书通过对大量程序实例的分析,深入浅出地讲解了 C/C++/C#语言基本知识、编译方法和程序设计技术,并给出了众多的编程技巧与应用。全书首先介绍了 C 语言的基本知识,对 C 语言编程的实际方法、技巧,以及常见的问题和难点进行了详细阐述和解答;然后逐步过渡到 C++编程与 Windows 编程,用较大的篇幅对它们进行了深入地剖析;最后介绍了基于 .NET 环境的 C#编程。无论是 C/C++/C#的初学者还是资深程序员都能从中受益匪浅。

本书是广大程序员、编程自学者学习和掌握 C/C++/C#编程的优秀指南。

First published by Delmar, a division of Thomson Learning.

All Rights Reserved. Authorized Simplified Chinese Edition by Thomson Learning and China WaterPower Press. No part of this book may be reproduced in any form without the express written permission of Thomson Learning and China WaterPower Press.

北京市版权局著作权合同登记号:图字 01-2002-1920 号

图书在版编目(CIP)数据

C/C++/C#程序员实用大全:精华版:C/C++/C#最佳编程指南/(美)耶姆塞(Jamsa,K.)著;张春晖等译.一北京:中国水利水电出版社,2005
(万水计算机技术实用大全系列)

书名原文:Jamsa's C/C++/C# Programmer's Bible The Ultimate Guide to C/C++/C# Programming

ISBN 7-5084-2892-7

I. C... II. ①耶…②张… III. C 语言—程序设计—指南 IV. TP312-62

中国版本图书馆 CIP 数据核字(2005)第 046634 号

书 名	C/C++/C#程序员实用大全——C/C++/C#最佳编程指南(精华版)
作 者	[美] Kris Jamsa 著
译 者	张春晖 刘大庆 李越 等译
审 校	利水文化
出版发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 54.25 印张 1260 千字
版 次	2005 年 5 月第 1 版 2005 年 5 月第 1 次印刷
印 数	0001—4000 册
定 价	78.00 元

凡购买我社图书,如有缺页、倒页、脱页者,本社营销中心负责调换

版权所有·侵权必究

精华版序

《万水计算机技术实用大全系列》自出版以来，受到了广大读者的欢迎和好评。在第一版售罄以后，仍有不少读者期望购买此套丛书，更有读者期望收藏此套丛书。为此我们征求了广大读者的意见，在第一版的基础上对整套丛书进行了改编，去掉了一些空洞的技巧，合并了罗嗦的章节段落，删除了陈旧的内容，使其更加符合中国读者的习惯，更加便于查阅。同时，从读者的角度考虑，我们在版面编排上略加压缩，在第一版的基础上缩减了图书的页面篇幅，让读者能以更低的价格获得更好的图书。

当今的计算机技术日新月异，软件的新版本不断推出，但其中除了技术本身的发展和 innovation 之外，还存在着软件供应商的商业行为。本套丛书囊括了 Visual Basic、SQL、Visual C++ 和 C/C++/C# 软件的各种实用技巧，是开发人员案头必备的一套参考丛书。当然，有人会说，部分软件已经有了新版本。从一方面考虑，众多读者还在使用原有版本，这套丛书非常适合于他们；而另一方面，即使读者采用了新版本的软件，但由于这些新旧版本软件所基于的底层技术是一样的，这套丛书同样也非常适合于他们。其实，读者的需求已经告诉我们，主流计算机软件的基本技术短期内不会改变，那么运用它们的基本技巧也不会随时间的流逝而淹没。这正是我们重新改编出版这套《万水计算机技术实用大全系列》精华版的初衷。我们相信，读者会喜欢这套丛书，也会从本套丛书中受益匪浅。

如果您是一名开发人员，那么一定很明白，软件需求千变万化，其代码实现也千变万化。在开发过程，经常会碰到为了解决某个问题而绞尽脑汁。您会疯狂地查阅网上资料或者图书资料，希望能够拓展思路或是获得具体实现方法，这时您多么渴望能有一本收集了各种实用技巧的图书！本套丛书的写作初衷就在于此，它融入了作者长期的开发经验，收集了相关软件的各种实用开发技巧。本套丛书最大的特点就是按软件功能给出一条条实用技巧，技巧之间相互引用，这样方便读者查阅。举例来说，如果您对 SQL 中的 BIT 数据类型不太了解，可以查阅《SQL 编程实用大全（精华版）》一书的目录，从第 1 章“理解 SQL 基本知识并创建数据库文件”中找到“技巧 16 理解 SQL 的 BIT 数据类型”，阅读该技巧即可获得相关知识。软件开发人员的工作时间往往都很长，如果有了这么一套丛书，当您冥思苦想某种实现方法时，不妨查阅一下本书，也许您可以多腾出几晚上的时间来喝咖啡，也许您能更多地赢得老板的赏识，让工作变得更轻松。

丛书精华版在改编过程中力求突出精华二字。首先是对上一版的文字进行了改编，使内容更简洁，更符合中文阅读习惯；其次删除了一些空洞的技巧、过于基础的内容、陈旧的内容、罕见的内容，或者是在新版本软件中已经不再采用的方法；最后丛书精华版在版面上进行了压缩，在较短的篇幅中容纳更多的内容，更适于作为参考书。本套丛书精华版适合于 Visual Basic、SQL、Visual C++ 和 C/C++/C# 开发人员作为参考书阅读和查阅，是开发人员解决实际问题 and 拓宽思路不可多得的一套丛书。

最后，感谢参与本套丛书改编的几位老师和开发人员，他们是互动出版网的姚先勤、哈尔滨理工大学的徐继伟、华北电力大学的燕士跃、北京英宇世纪信息技术有限公司的李秋霞。其中部分内容融入了他们的开发实践，在此谨表感谢！

译者序

自从本书第一版的中文版正式面世以来，受到广大读者的好评，已重印数次。本书第二版在第一版的基础上增加了有关.NET 环境和 C#编程的知识，并进一步完善和丰富了各部分的内容，使读者能够通过一本书就深入掌握 C/C++/C#编程。

在本书的翻译过程中，我们常常为作者对 C/C++/C#编程的深入掌握和独到见解而惊讶、赞叹，对我们自己而言，这也是一个学习与提高的过程，同时也为能向国内读者推荐这本好书而高兴。通过认真阅读这本优秀的 C/C++/C#编程指南，读者将逐渐成为 C/C++/C#的编程大师。全书先后讲述了 C 语言编程、C++语言编程、Windows 编程的知识和 C#语言编程。

C 语言编程

近年来，C 语言无论在国外还是在国内都得到了迅速地推广。C 语言以它的功能丰富、表达能力强、使用灵活、应用面广、目标程序效率高、可移植性好、能对硬件直接进行操作等优点愈来愈赢得人们的青睐。

C 语言功能强、使用灵活，但是普遍认为它比较难学，学习和使用 C 语言需要有一定的硬件和软件基础知识，不熟练的程序员常常出错而不知其所以然。针对这种情况，本书从 C 语言的基础知识开始，然后讲述了 C 语言中的宏与常量、各种变量类型、函数、键盘操作、文件与磁盘操作、各种构造类型的数据结构、DOS 和 BIOS 服务、内存管理、重定向 I/O、编程工具、高级 C 语言编程等知识，使读者很快地掌握 C 语言的核心技术进而全面、深入地掌握 C 语言编程的奥妙之处。

C++语言编程

面向对象的编程是当前的热门课题，C++语言作为 C 语言的超集，对面向对象的程序设计提供了强有力的支持，受到了越来越多的程序设计员的青睐。但是面对一大堆 C++程序员手册、编程指南等参考资料，学习了 C++语言和面向对象程序设计的思想，却不知如何下手编程，无法用类来实现真实的模型。甚至有些程序员仍然用结构化的程序设计思想来编写 C++程序，有些程序员设计的类及类构造明显违背了面向对象程序设计的原则，没有发挥出 C++潜在的威力。本书正是基于上述考虑，为培养最优秀的 C++程序员而编写的，所讲述的编程技巧是作者丰富经验和体会的结晶。主要内容有：I/O 流和可流类的奥秘、函数模板和类模板的探讨、对象构造和对象析构方面的技术，并详细叙述了基于类模板的标准模板类库的使用方法。本书是一本实用手册，它用许多程序和工程实例论述了 C++编程的精髓，初学者可以从书中了解到面向对象程序设计的思想；通过套用现成的编程方法，编写出合乎面向对象程序设计原则的程序，少走弯路。对于已有一定 C++编程经验的读者，

本书会使他们对 C++ 特性及实现有更深地理解，在编程中更充分地发挥 C++ 的潜在威力。

Windows 编程

Windows 的 32 位操作系统增加了许多新的功能和特性，特别是 32 位的寻址能力完全打破了原来的 640KB 段寻址方式的限制，所以大大提高了应用程序的运行效率和运行速度。本书除了介绍 Windows 的基本编程技巧外，还着重介绍了 32 位操作系统提供的新功能的编程技巧，内存管理、多线程、异步 I/O 等的编程技巧。本书内容丰富，例证充实，每一个技巧都给出了编程人员最关心的问题，并进行了详细的说明和演示，而且所涉及的内容相当广泛，编程人员几乎可以在本书中找到他想利用的任何 Windows 编程主题，通过本书可以使以前感到非常神秘而难以理解的问题，现在看来是那么自然。本书使读者能充分发挥 32 位操作系统的威力。

C# 语言编程

C#(读音为 C sharp)作为 Microsoft 公司集成开发环境 Visual Studio .NET 的一个组件，是一种完全面向对象的语言。.NET 的核心开发环境是 Visual Studio .NET，它用大家熟悉的工具为开发者提供了单一的编程环境。.NET 框架的很大一部分都是用 C# 编写的。C# 是一种高级编程语言，它兼容了 17 种编程语言的优点，尤其是加入了其主要竞争对手 Sun 公司的产品 Java 语言的许多优点。按照 Microsoft 的说法，C# 将取代其他高级编程语言，成为主导编程语言。相信读者通过阅读本书的 C# 编程部分之后，会对 C# 有很深刻的印象，喜欢上它，并使用它开发新的应用程序。

本书主要由张春晖和刘大庆负责翻译，利水文化审校。参加本书翻译工作的还有：王可欣、李越、王军、张相文、谢衍球、张春玉、张春霞、晏飞、韩素华、周春元、牛彦斌、方瑾、东纯海、金学东、周刚、赵玉琰等。本书翻译过程中得到了孙春亮的大力帮助与支持，在此表示感谢！

由于译者水平有限，错误之处在所难免，恳请广大读者批评指正。

译者
2002 年 6 月

目 录

精华版序

译者序

第一章 C 语言入门.....	1
1 典型的 C 程序结构.....	1
2 C 语言区分大小写字母.....	1
3 帮助编译器查找头文件.....	1
4 注释程序.....	2
5 变量.....	3
6 给变量赋值.....	3
7 变量类型.....	4
8 定义同一类型的多个变量.....	4
9 给变量赋初值.....	4
10 C 关键字.....	4
11 整型变量.....	5
12 字符型变量.....	5
13 浮点型变量.....	6
14 双精度型变量.....	6
15 给浮点型变量赋值.....	6
16 类型标识符.....	7
17 联合使用无符号和长类型标识符.....	7
18 寄存器类型标识符.....	7
19 有符号类型标识符.....	8
20 创建用户自己的类型.....	8
21 赋给十六进制或八进制值.....	9
22 溢出.....	9
23 赋值为引号或其他字符.....	9
24 PRINTF 入门.....	10
25 打印八进制或十六进制整数.....	10
26 用 PRINTF 显示无符号整型数值.....	11
27 用 PRINTF 显示长整型数值.....	11
28 用指数格式显示浮点数.....	12
29 显示浮点数.....	12
30 在数值前添加正号和负号.....	12
31 用 PRINTF 格式化整数值.....	13
32 0 填充整数输出.....	13
33 在八进制和十六进制数前显示前缀.....	14

34 用 PRINTF 格式化浮点数.....	14
35 格式化指数输出.....	14
36 左对齐 PRINTF 的输出.....	15
37 显示 NEAR 和 FAR 字符.....	15
38 使用 PRINTF 的转义字符.....	16
39 使用 PRINTF 的返回值.....	16
40 使用 ANSI 设备驱动器.....	17
41 用 ANSI 驱动器显示屏幕颜色.....	17
42 用 ANSI 驱动器定位光标.....	18
43 在 C 中作基本的数学运算.....	18
44 模运算（取余运算）.....	19
45 运算符的优先级和结合性.....	19
46 C 的自增运算符.....	20
47 按位或运算.....	21
48 按位与运算.....	21
49 按位异或运算.....	22
50 “取反”运算.....	23
51 C 的条件运算符.....	23
52 C 的长度（SIZEOF）运算符.....	24
53 移位运算.....	24
54 位循环运算.....	25
55 C 如何表示真（TRUE）和假 （FALSE）.....	26
56 用 if 判断条件.....	26
57 判断是否相等.....	26
58 关系判断.....	27
59 用逻辑与判断两个条件.....	27
60 用逻辑或来判断两个条件.....	28
61 逻辑非运算.....	28
62 将条件结果赋值给变量.....	28
63 判断浮点数.....	29
64 按规定次数执行语句.....	29
65 无穷循环.....	30
66 在 FOR 循环中使用逗号运算符.....	30

67	用 WHILE 循环重复执行一条或多条语句.....	31
68	使用 DO 重复执行一条或多条语句.....	32
69	C 的 CONTINUE 语句.....	33
70	使用 C 的 BREAK 语句来结束循环.....	34
71	GOTO 语句分支.....	34
72	判断多个条件.....	35
73	在 SWITCH 中使用 BREAK.....	36
74	使用 SWITCH 语句的 DEFAULT CASE.....	37
第二章 宏与常量.....39		
75	在程序中定义常量.....	39
76	使用 _FILE_ 预处理器常量.....	39
77	使用 _LINE_ 预处理器常量.....	40
78	改变预处理器的行计数.....	40
79	记录预处理器的日期和时间.....	41
80	判断是否进行 ANSI C 编译.....	41
81	判断是 C++ 还是 C.....	41
82	取消宏或常量.....	42
83	比较宏与函数.....	42
84	编译器 PRAGMAS.....	42
85	使用 #INCLUDE <FILENAME.H> 与 #INCLUDE "FILENAME.H".....	42
86	判断符号是否被定义.....	43
87	进行 IF-ELSE 预处理.....	43
88	更强大的预处理器条件判断.....	44
89	实现 IF-ELSE 和 ELSE-IF 预处理.....	44
90	定义需要多行的宏和常量.....	45
91	创建自定义宏.....	45
92	创建 MIN 和 MAX 宏.....	46
93	创建 SQUARE 和 CUBE 宏.....	46
94	注意宏定义中的空格.....	46
95	如何使用括号.....	47
96	宏是没有类型的.....	47
第三章 字符串.....49		
97	C 字符串的形象化.....	49
98	C 是如何存储字符串的.....	49
99	在字符串常量内表示引号.....	50
100	使用 STRLEN 函数.....	50
101	字符串的复制.....	50
102	字符串的连接.....	51
103	给字符串追加 N 个字符.....	51
104	判断两个字符串是否相同.....	51
105	比较字符串时忽略大小写.....	52
106	将字符串转换成大写或小写.....	53
107	获取字符串中第一次出现的某个字符.....	53
108	搜索字符在字符串中的末次出现.....	54
109	计算字符串中某个字符出现的次数.....	54
110	比较两个字符串.....	54
111	比较两个字符串的前 N 个字符.....	55
112	不考虑大小写比较字符串.....	55
113	将字符串转换成数字.....	56
114	复制字符串的内容.....	56
115	在字符串中查找子字符串.....	57
116	计算子字符串出现的次数.....	57
117	获取子字符串的最右端出现.....	58
118	从字符串中删除子字符串.....	58
119	用另一个子字符串代替子字符串.....	58
120	转换数值的 ASCII 码形式.....	59
121	判断字符是否为字母或数字.....	60
122	判断字符是否为字母.....	60
123	判断字符是否包含 ASCII 值.....	60
124	判断字符是否为控制符.....	60
125	判断字符是否为数字.....	61
126	判断字符是否为图形字符.....	61
127	判断字符是大写还是小写.....	61
128	判断字符是否可打印.....	61
129	判断字符是否为标点符号.....	62
130	判断字符是否包含空白符 (whitespace).....	62
131	判断字符是否为十六进制值.....	62
132	将字符转换成大写形式.....	62
133	使用 ASCII 字符.....	63
134	将输出格式写进字符串变量.....	63
135	从字符串中读输入.....	64
136	标志字符串以节省空间.....	64
137	初始化字符串.....	64
第四章 函数.....66		
138	函数.....	66
139	把 MAIN 当作函数.....	66

140	函数返回值.....	67
141	形参和实参.....	68
142	解决名称冲突.....	68
143	局部变量.....	68
144	函数如何使用堆栈.....	69
145	C 如何存储局部变量.....	69
146	声明全局变量.....	69
147	传值调用.....	70
148	使用传值调用防止参数值变化.....	71
149	传址调用.....	71
150	获取地址.....	72
151	使用变量的地址.....	72
152	改变参数的值.....	73
153	只改变指定参数.....	74
154	使用堆栈进行传址调用.....	74
155	记住函数变量的值.....	75
156	C 是如何初始化静态变量的.....	76
157	使用 PASCAL 调用顺序.....	76
158	PASCAL 关键字的影响.....	76
159	混合编程示例.....	77
160	CDECL 关键字.....	77
161	判断是否要使用递归.....	78
162	如何消除递归.....	79
163	形式参数中的 CONST.....	79
164	使用 CONST 不会阻止参数值的修改..	80
165	无界字符串的声明.....	80
166	C 是如何使用堆栈处理字符串参数的....	81
167	外部变量.....	81
168	外部静态变量.....	82
169	VOLATILE 关键字.....	83
170	调用结构和基指针.....	83
171	调用汇编语言函数.....	83
172	从汇编语言函数中返回值.....	84
173	AUTO 关键字.....	85
174	支持参数个数可变的函数.....	85
175	VA_START、VA_ARG 和 VA_END 是如何工作的.....	86
176	创建支持多参数多类型的函数.....	87
第五章 键盘操作.....		89
177	从键盘读入字符.....	89

178	显示字符输出.....	89
179	将键盘输入赋给字符串.....	89
180	使用直接 I/O 读入字符.....	90
181	不显示字符的直接键盘输入.....	90
182	直接输出.....	91
183	将按键放回键盘缓存.....	92
184	使用 CPRINTF 快速格式化输出.....	92
185	快速格式化键盘输入.....	93
186	写字符串.....	93
187	使用直接 I/O 实现更快的字符串输出....	94
188	从键盘读入字符串.....	95
189	以更快的速度从键盘输入字符串.....	95
190	清除屏幕显示.....	95
191	删除当前行到行尾的内容.....	96
192	删除屏幕上的当前行.....	96
193	定位光标进行屏幕输出.....	96
194	判断行与列的位置.....	97
195	在屏幕上插入空行.....	97
196	将屏幕上的文本拷贝到缓冲区.....	97
197	将缓冲区中的文本拷贝到屏幕的 指定位置.....	98
198	判断文本模式设置.....	99
199	控制屏幕颜色.....	99
200	指定背景色.....	100
201	使用 TEXTCOLOR 设置前景色.....	101
202	使用 TEXTBACKGROUND 设置 背景色.....	102
203	控制文本的明暗度.....	102
204	决定当前文本模式.....	103
205	在屏幕上移动文本.....	103
206	定义文本窗口.....	104
第六章 数学.....		105
207	使用整型表达式的绝对值.....	105
208	使用 ARCCOSINE (反余弦).....	105
209	使用 ARCSINE (反正弦).....	105
210	使用 ARCTANGENT (反正切).....	106
211	求复数的绝对值.....	106
212	对浮点值进位舍入.....	107
213	使用角的余弦.....	107
214	使用角的正弦.....	107

215	使用角的双曲正弦.....	108	254	刷新 I/O 流.....	129
216	使用角的正切.....	108	255	一次关闭所有被打开的文件.....	130
217	使用角的双曲正切.....	109	256	使用 P_TMPDIR 创建临时文件名.....	130
218	整数相除.....	109	257	使用 TMP 或 TEMP 创建临时文件名.....	130
219	使用指数.....	109	258	创建真正的临时文件.....	131
220	使用浮点型表达式的绝对值.....	110	259	删除临时文件.....	131
221	使用浮点余数.....	110	260	为文件搜索命令路径.....	131
222	使用浮点值的尾数和指数.....	110	261	为文件搜索环境入口的子目录.....	132
223	计算自然对数.....	111	262	打开 TEMP 目录中的文件.....	132
224	判断最大值与最小值.....	111	263	最小化文件 I/O 操作.....	133
225	把浮点值分解成整数和小数部分.....	112	264	改变当前目录.....	134
226	计算 x^n 的结果.....	112	265	创建目录.....	134
227	生成随机数.....	113	266	删除目录.....	134
228	将随机值映射到指定范围.....	113	267	删除目录树.....	135
229	给随机数生成器赋初值.....	114	268	建立完全路径名.....	135
230	计算数值的平方根.....	114	269	分解目录路径.....	135
231	创建定制的数学错误处理程序.....	115	270	建立路径名.....	136
第七章	文件、目录和磁盘.....	117	271	使用低级函数打开和关闭文件.....	137
232	判断当前盘驱动器.....	117	272	创建文件.....	138
233	选择当前驱动器.....	117	273	进行低级读写操作.....	139
234	判断可用的盘空间.....	117	274	为文件句柄翻译指定模式.....	139
235	读入文件分配表 (FAT) 信息.....	118	275	利用 LSEEK 定位文件指针.....	139
236	绝对扇区读写操作.....	119	276	使用 DOS 文件服务.....	140
237	进行 BIOS 磁盘 I/O.....	119	277	获取文件的日期和时间标记.....	141
238	测试软驱是否准备好.....	121	278	设置文件的日期与时间标记.....	142
239	应用 FOPEN 打开文件.....	121	279	把文件日期和时间设置成当前日期 和时间.....	143
240	FILE 结构.....	122	280	每次读写一个字.....	143
241	关闭一个打开的文件.....	122	281	改变文件的长度.....	144
242	每次读 / 写文件信息的一个字符.....	123	282	控制文件打开操作的读写模式.....	145
243	文件指针的位置指针.....	123	283	将缓冲区赋给文件.....	146
244	判断当前文件位置.....	124	284	分配文件缓冲区.....	146
245	文件翻译.....	124	285	利用 MKTEMP 创建惟一文件名.....	147
246	文件句柄 (FILE HANDLES).....	124	286	读写结构.....	148
247	从流指针中导出文件句柄.....	125	287	从文件流中读取结构数据.....	149
248	重命名文件.....	125	288	复制文件句柄.....	150
249	删除文件.....	126	289	强制文件句柄设置.....	150
250	判断程序如何访问文件.....	126	290	把文件句柄和文件流联系起来.....	151
251	设置文件的访问模式.....	127	291	文件共享.....	151
252	检测文件流错误.....	128	292	打开文件进行共享访问.....	152
253	判断文件的长度.....	129			

293	锁定文件内容.....	152	332	利用 BSEARCH 搜索已排序数组.....	177
294	获取更精细的文件锁定控制.....	154	333	利用 QSORT 对数组排序.....	178
295	使用 DOS 目录.....	155	334	判断数组元素的个数.....	179
296	打开目录.....	155	335	判断变量的地址.....	179
297	递归读取磁盘文件.....	155	336	声明指针变量.....	180
298	判断当前文件位置.....	156	337	间接访问指针.....	180
299	打开共享文件流.....	157	338	指针与函数参数的使用.....	181
300	在指定目录中创建惟一文件.....	157	339	指针运算.....	181
301	创建新文件.....	158	340	指针的增值与减值.....	181
302	强制二进制或文本文件打开.....	159	341	利用指针遍历数组.....	182
303	按行读文本.....	159	342	使用返回值为指针的函数.....	182
304	按行写文本.....	159	343	创建返回值为指针的函数.....	182
305	应用 FGETS 和 FPUTS.....	160	344	指针数组.....	183
306	强制二进制文件翻译.....	160	345	遍历字符串数组.....	183
307	判断文件结尾.....	160	346	把字符串数组当成指针.....	184
308	舍弃字符.....	161	347	使用指向一个指向字符串的指针 的指针.....	184
309	读取格式化的文件数据.....	161	348	利用指针声明字符串常量.....	184
310	根据当前位置定位文件指针.....	162	349	VOID 类型指针.....	185
311	获取文件句柄信息.....	162	350	创建指向函数的指针.....	185
312	重新打开文件流.....	163	351	使用指向函数的指针.....	185
第八章	数组、指针和结构.....	164	352	使用三级指针.....	186
313	声明数组.....	164	353	结构.....	186
314	初始化数组.....	164	354	用不同的方式声明结构.....	187
315	使用常量定义数组.....	164	355	结构成员.....	187
316	为大数组使用 HUGE 内存模式.....	165	356	形象表示结构.....	187
317	多维数组.....	166	357	应用结构.....	188
318	给二维数组元素赋初值.....	166	358	把结构传递给函数.....	188
319	把多维数组当作一维数组.....	166	359	在函数内部改变结构.....	189
320	联合.....	167	360	(*POINT).MEMBER 间接引用.....	189
321	使用联合节省内存.....	168	361	使用 POINTER→MEMBER 格式.....	190
322	使用 REGS —— 一种典型的联合.....	168	362	初始化结构.....	191
323	应用 REGS 联合.....	169	363	使用嵌套结构.....	191
324	位字段结构.....	170	364	包含数组的结构.....	191
325	在数组中查找指定的值.....	171	365	创建结构数组.....	192
326	冒泡排序法.....	172	第九章	DOS 和 BIOS 服务.....	193
327	选择排序法.....	173	366	DOS 系统服务.....	193
328	SHELL (希尔) 排序法.....	173	367	BIOS 服务.....	193
329	快速排序法.....	175	368	寄存器.....	194
330	利用 LFINF 搜索字符串.....	176	369	标志寄存器.....	195
331	利用 LSEARCH 搜索数值.....	177			

370	软件中断.....	196	408	分配动态内存.....	219
371	利用 BIOS 访问指针.....	196	409	再谈类型转换.....	220
372	CONTROL+BREAK 信息.....	198	410	释放内存.....	220
373	可能的 DOS 副作用.....	198	411	利用 CALLOC 函数分配内存.....	220
374	暂时挂起程序.....	199	412	堆.....	221
375	控制声音.....	199	413	解决 64Kb 堆限制.....	222
376	获取国家专用的信息.....	200	414	从堆栈中分配内存.....	223
377	磁盘传输地址.....	201	415	分配巨型数据.....	224
378	访问和控制磁盘传输区.....	201	416	改变被分配内存区域的大小.....	224
379	BIOS 键盘服务.....	202	417	BRK 函数.....	225
380	获取 BIOS 设备列表.....	203	418	快速堆检测.....	225
381	控制串行口 I/O.....	204	419	填充自由堆空间.....	226
382	利用 BDOS 访问 DOS 服务.....	206	420	检测特定的堆入口.....	227
383	获取扩展 DOS 错误信息.....	207	421	遍历堆入口.....	227
384	判断 BIOS 常规内存数量.....	208	422	访问指定内存单元.....	228
385	创建远指针 (FAR PRINTER).....	208	423	向内存中置数.....	229
386	把远端地址分解为段地址和偏移 地址.....	209	424	PC 端口.....	229
387	判断自由核心内存.....	209	425	访问端口值.....	230
388	读段寄存器设置.....	209	426	CMOS.....	230
第十章	内存管理.....	211	427	内存模式.....	231
389	内存的类型.....	211	428	微型内存模式.....	231
390	常规内存.....	211	429	小型内存模式.....	232
391	常规内存的分布.....	211	430	中型内存模式.....	232
392	为什么 PC 和 DOS 限制于 1MB.....	212	431	压缩内存模式.....	233
393	从段和偏移量中产生地址.....	212	432	大型内存模式.....	234
394	扩充内存.....	213	433	巨型内存模式.....	235
395	使用扩充内存.....	213	434	判断当前的内存模式.....	235
396	实模式和保护模式.....	214	第十一章	日期和时间.....	236
397	访问扩展内存.....	214	435	获取当前日期与时间.....	236
398	高端内存区.....	215	436	延迟若干毫秒.....	236
399	堆栈.....	215	437	判断程序的耗时.....	236
400	各种堆栈配置.....	215	438	比较两个时间.....	237
401	判断程序的当前堆栈大小.....	216	439	获取日期数据串.....	237
402	使用_STKLEN 控制堆栈空间.....	216	440	获取时间串.....	238
403	给内存区域赋值.....	217	441	读 BIOS 计时器.....	238
404	拷贝内存区域.....	217	442	使用当地时间.....	239
405	拷贝内存区域直到某指定字节.....	217	443	使用格林威治平时.....	239
406	比较两个无符号字符数组.....	218	444	获取 DOS 系统时间.....	240
407	交换两个相邻字符串字节.....	218	445	获取系统日期.....	240
			446	设置 DOS 系统时间.....	241

447	设置 DOS 系统日期.....	241
448	把 DOS 日期转换为 UNIX 格式.....	241
449	利用 TIMEZONE 计算时差.....	242
450	判断当前时区.....	242
451	利用 TZSET 设置时区区域.....	242
452	利用 TZ 环境入口.....	243
453	从用户程序中设置 TZ 环境入口.....	243
454	获取时区信息.....	243
455	以秒钟的形式设置自 1/1/1970 午夜 以来的系统时间.....	244
456	把日期转换成自 1/1/1970 以来的 秒数.....	244
457	判断日期的儒略历日期.....	245
458	创建格式化日期和时间串.....	246
459	PC 时钟类型.....	247
第十二章 重定向 I/O 和进程命令行.....		248
460	等候按键.....	248
461	提醒用户输入密码.....	248
462	自己编写密码函数.....	248
463	输出重定向.....	249
464	输入重定向.....	250
465	联合使用 INPUT 和 OUTPUT 重 定向.....	250
466	利用 STDOUT 和 STDIN.....	250
467	管道 (pipe) 运算符.....	251
468	GETCHAR 和 PUTCHAR.....	251
469	确保信息出现在屏幕上.....	251
470	自定义 MORE 命令.....	252
471	防止 I/O 重定向.....	253
472	应用 STDPN 文件句柄.....	254
473	把重定向输出分割到一个文件中.....	254
474	应用 STDAUX 文件句柄.....	255
475	命令行变元.....	255
476	从命令行中显示文件内容.....	256
477	把 ARGV 当作指针.....	256
478	C 是如何知道命令行的.....	257
479	环境.....	257
480	把 ENV 当作一个指针.....	258
481	使用命令行数字.....	258
482	出口状态值.....	258

483	为出口状态过程使用 RETURN.....	259
484	在环境中搜索特定入口.....	260
485	DOS 是如何对待环境的.....	260
486	应用 ENVIRON 全局变量.....	261
487	给当前环境添加入口.....	262
488	给 DOS 环境添加元素.....	262
489	退出当前程序.....	263
490	定义在程序结束时执行的函数.....	263
第十三章 编程工具.....		264
491	库.....	264
492	重复使用目标代码.....	264
493	创建库文件.....	265
494	常用的库操作.....	265
495	列出库文件中的子例行程序.....	265
496	库管理程序的其他功能.....	265
497	连接器.....	266
498	连接器的功能.....	266
499	使用连接映像.....	266
500	使用连接器响应文件.....	267
501	使用 MAKE 命令简化应用程序的 创建.....	267
502	生成一个简单的 MAKE 文件.....	268
503	通过 MAKE 使用多依赖性文件.....	268
504	说明用户的 MAKE 文件.....	269
505	MAKE 和命令行.....	269
506	在 MAKE 文件中放置多个依赖性.....	270
507	显现的和隐含的 MAKE 法则.....	270
508	使用 MAKE 宏.....	270
509	预定义 MAKE 宏.....	271
510	用 MAKE 执行条件进程.....	271
511	验证一个 MAKE 宏.....	272
512	再次包含一个 MAKE 文件.....	272
513	使用 MAKE 的宏修饰符.....	272
514	因错误结束 MAKE 文件.....	273
515	关闭命令显示.....	273
516	使用文件 BUILTINS.MAK.....	273
517	在 MAKE 中执行出口状态进程.....	273
518	同时激活和改变一个宏.....	273
519	为多个依赖文件执行一个 MAKE 命令.....	274

第十四章 高级 C 语言编程	275	559 生成一个中断.....	298
520 判断是否有数学协处理器.....	275	560 捕获 PC 机时钟.....	298
521 理解 CTYPE.H, ICTYPE 宏.....	275	561 C 语言中的致命错误处理.....	299
522 检查系统和路径错误.....	276	562 一个更完全的致命错误处理器.....	300
523 显示预定义的错误信息.....	277	563 恢复改变过的中断.....	301
524 决定操作系统版本.....	277	564 生成一个 Ctrl+Break 处理器.....	301
525 执行一个非本地的 GOTO.....	277	565 在用户的致命错误处理器中使用 DOS 服务.....	302
526 获得进程 ID (PID)	278	566 直接插入内联函数.....	302
527 激活一个内部的 DOS 命令.....	278	567 使能和禁止内联函数.....	303
528 使用 _PSP 全局变量.....	279	568 理解快速函数调用.....	303
529 在变量声明中使用 CONST 修饰符... ..	280	569 _FASTCALL 参数传递的法则.....	304
530 使用枚举类型.....	280	570 消除普通的子表达式.....	304
531 分配一个特殊的值给枚举类型.....	281	571 理解初始化值.....	304
532 保存和恢复寄存器.....	281	572 理解连接.....	305
533 动态列表简介.....	281	573 理解左值 (LVALUE)	305
534 声明一个链接的列表结构.....	282	574 理解右值 (RVALUE)	306
535 增加一个列表入口.....	283	575 使用段寄存器关键字.....	306
536 插入一个列表入口.....	283	576 谨慎使用远指针.....	306
537 从一个列表中删除一个元素.....	283	577 数学协处理器语句.....	307
538 使用一个双向链表.....	284	578 理解变量中的 CDECL 和 PASCAL... ..	307
539 创建一个简单的双向链表.....	285	579 防止循环包含.....	307
540 理解 NODE->PREVIOUS->NEXT	285	第十五章 C++入门	308
541 从一个双向链表中移走一个元素.....	286	580 C++介绍.....	308
542 在一个双向链表中插入一个元素.....	287	581 理解 COUT I/O 流.....	308
543 理解子进程.....	288	582 使用 COUT 输出值和变量.....	308
544 派生一个子进程.....	288	583 显示十六进制和八进制数值.....	309
545 使用其他的 SPAWNLXX 函数.....	289	584 输出到 CERR.....	309
546 使用 SPAWNVXX 函数.....	290	585 用 CIN 得到输入.....	309
547 执行一个子进程.....	291	586 使用 CLOG 实现输出.....	309
548 使用其他的 EXECLXX 函数.....	292	587 CIN、COUT、CERR 和 CLOG 是 类的实例.....	310
549 使用 EXECVXX 函数.....	292	588 使用 FLUSH 操纵符快速输出.....	310
550 理解覆盖.....	293	589 C++增加的新关键字.....	310
551 理解中断.....	293	590 分辨全局范围.....	310
552 PC 机的中断.....	294	591 提供缺省参数值.....	311
553 使用中断关键字.....	295	592 控制 COUT 的输出宽度.....	311
554 判断一个中断向量.....	295	593 使用 SETW 设置 COUT 宽度.....	311
555 设置一个中断向量.....	296	594 指定 COUT 的填充字符.....	312
556 使能与禁止中断.....	296	595 左对齐和右对齐 COUT 的输出.....	312
557 生成简单的中断处理器.....	296		
558 链接一个二次中断.....	297		

596	控制 COUT 显示浮点数字的数目	313
597	以小数或科学记数格式显示数值	314
598	恢复 COUT 至默认值	314
599	设置输入输出基数	314
600	在需要的地方定义变量	315
601	在函数原型中放置默认参数值	315
602	使用按位运算符及 COUT	316
603	在 C++ 中使用 ENUM 关键字	316
604	用 NEW 分配内存	316
605	使用 FAR 指针和 NEW 运算符	316
606	释放内存至自由空间	317
607	理解 C++ 中的引用	317
608	给函数传递引用	318
609	防止隐藏对象	318
610	用三种方法传递参数	318
611	使用引用的规则	319
612	函数可返回引用	319
613	使用 INLINE 关键字	320
614	使用 C++ 的 ASM 关键字	321
615	用 CIN 读字符	321
616	用 COUT 写字符	321
617	文件结束测试	322
618	用 ENDL 产生新行	322
619	理解连接规范	322
620	理解重载	323
621	重载函数	323
622	重载函数的第二个例子	324
623	避免不明确的重载	324
624	使用 CIN 每次读一行	325
625	改变 NEW 运算符的缺省处理器	325
626	用 SET_NEW_HANDLER 函数设置 NEW 处理器	326
627	判断 C++ 编译	326
628	理解 C++ 中的结构	327
629	结构中定义函数成员	327
630	在结构内定义成员函数	328
631	在结构外定义成员函数	328
632	给成员函数传递参数	328
633	不同结构具有同名函数成员	329
634	同名成员不同函数	329

第十六章 对象	331
635 对象	331
636 把程序分解成对象	331
637 理解 C++ 的类	333
638 理解封装	334
639 理解多态性	334
640 理解继承	334
641 类和结构的选择	334
642 创建简单类模型	335
643 实现简单类的程序	335
644 定义类构件	336
645 理解作用域分辨符	337
646 在声明中使用或省略类名	337
647 理解 PUBLIC: 标志	337
648 理解 PRIVATE: 标志	338
649 理解 PROTECTED: 标志	338
650 公有方法常称为接口函数	338
651 对象实例共享代码	339
652 全局分辨符	339
653 类初始化	339
654 另一种类初始化方法	340
655 理解静态类成员	340
656 静态数据成员	341
657 静态成员函数	342
658 成员函数声明	342
659 内联函数声明	343
660 决定内联和非内联函数的使用	343
661 类和联合	344
662 匿名联合	344
663 友元函数	345
第十七章 常见的类函数	346
664 构造函数	346
665 使用构造函数	346
666 构造函数的缺省参数	347
667 重载构造函数	348
668 获得重载函数的地址	349
669 析构函数	349
670 程序何时调用析构函数	350
671 使用复制构造函数	352
672 显式构造函数	353

673	局部类.....	353	711	用 WRITE 写二进制数据	379
674	构造函数和类数组	354	712	使用 GCOUNT 成员函数	380
675	运算符重载.....	354	713	使用重载的 GET 函数	381
676	创建运算符成员函数.....	355	714	使用 GETLINE 方法	381
677	重载加运算符	355	715	文件结束的检测.....	382
678	重载前置和后置增量运算符	356	716	使用 IGNORE 函数.....	383
679	用友元函数重载运算符	357	717	使用 PEEK 函数.....	384
680	用友元函数重载++或--运算符	358	718	使用 PUTBACK 函数	384
681	用友元函数重载运算符的原因	359	719	获得文件流的当前位置	384
682	重载 NEW 运算符	360	720	控制文件流指针	384
683	重载 DELETE 运算符	360	721	用 SEEKG 和 SEEKP 实现随机访问 ..	385
684	重载数组下标运算符[]	362	722	在文件内操纵文件指针位置	385
685	重载函数调用运算符()	363	723	判断 I/O 流的当前状态.....	386
686	重载指针运算符->	364	724	字符串流.....	387
687	重载逗号运算符	364	725	用 ISTRSTREAM 写字符串	387
688	取消输入前的空格	366	726	对输出数组使用 PCOUNT	388
689	在头文件中定义类.....	366	727	用 IOS 成员函数操纵流数组	388
690	创建初始化的类数组	366	728	使用 STRSTREAM	389
691	内联类代码允许改变.....	367	729	随机访问流数组.....	389
692	静态存储区.....	368	730	对流数组使用操纵符.....	390
第十八章 用 C++输入输出		369	731	为字符串数组定制插入运算符.....	390
693	I/O 流操作和 STDIO 同步.....	369	732	I/O 流中使用动态数组	392
694	用 IOS 成员格式化输入和输出	369	733	ENDS 操纵符	393
695	设置格式标志	370	734	对象调用另一对象	393
696	清除格式标志	370	735	重温友元.....	394
697	重载 SETF 函数	370	736	声明 READER 类为友元.....	394
698	检测当前格式标志	371	737	消除 CLASS 类名语句	395
699	使用 PRECISION 函数	372	738	限制友元访问.....	395
700	使用 FILL 函数	372	739	名字冲突及友元.....	396
701	理解操纵算子	373	第十九章 继承及多态性		397
702	用操纵算子格式化 I/O	373	740	C++中的继承.....	397
703	重载抽取运算符	374	741	基类和派生类.....	398
704	重载 COUT 的插入运算符的另一种 方法	375	742	派生类.....	398
705	打开文件流.....	375	743	基类及派生类的构造函数.....	398
706	读和写文件流数据.....	376	744	使用受保护的成员	399
707	把文件流操作放到一起	376	745	公有的和私有的基类继承.....	400
708	执行二进制拷贝操作	377	746	受保护的基类继承	401
709	流缓冲类.....	378	747	多重继承.....	402
710	用 READ 读二进制数据	378	748	构造函数的顺序和基类	403
			749	声明私有基类.....	404

750	解决类及其基类名字的冲突.....	405	787	用通用函数压缩数组.....	433
751	如何为基类构造函数传递参数.....	405	788	模板同样消除类复制.....	434
752	派生类内的访问声明.....	407	789	通用类.....	435
753	用虚拟基类避免混淆.....	407	790	创建含两个通用数据类型的通用类..	436
754	虚拟基类.....	408	791	创建含有参数的操纵符.....	437
755	派生类使用保护成员.....	409	792	通用数组类.....	438
756	定义静态类数据.....	410	第二十一章 异常处理及类型转换.....		440
757	初始化静态数据成员.....	411	793	异常处理.....	440
758	直接访问静态数据成员.....	412	794	基本的异常处理形式.....	440
759	静态私有数据成员.....	412	795	简单的异常处理器.....	440
760	静态成员函数.....	413	796	THROW 语句.....	441
761	公有静态函数的直接访问.....	413	797	异常具有特定类型.....	441
762	嵌套类.....	414	798	TRY 块内的函数抛出异常.....	442
763	在方法函数中插入汇编语句.....	415	799	函数内的局部 TRY 块.....	442
764	类成员可递归.....	415	800	程序何时执行 CATCH.....	443
765	THIS 指针.....	416	801	对单个 TRY 块使用多个 CATCH	
766	指向类的指针.....	417	语句.....		444
767	对不同的类使用相同的指针.....	418	802	对异常使用省略符 (···)	445
768	使用指针时基类和派生类名字的		803	捕获单个 TRY 块的所有异常.....	445
冲突		418	804	捕获单个 TRY 块的显式和通用异常.....	446
769	继承虚拟属性.....	419	805	异常的限制.....	447
770	虚拟函数是分层的.....	420	806	重新抛出异常.....	448
771	实现多态性.....	421	807	异常处理的应用.....	449
772	纯虚函数.....	422	808	使用缺省的函数参数.....	449
773	使用虚拟函数.....	422	809	用缺省函数参数避免错误.....	450
774	先期和迟后联编演示.....	424	810	缺省参数和函数重载.....	450
775	定义输出流操纵符.....	425	811	创建转换函数.....	451
776	对类使用 SIZEOF.....	425	812	使用转换函数提高类型的可移植性..	452
777	PRIVATE、PUBLIC、PROTECTED		813	C++中新的强制转换运算符.....	453
也能用于结构.....		426	814	CONST_CAST 运算符.....	453
778	友元转换函数.....	426	815	DYNAMIC_CAST 运算符.....	454
779	判断运算符提高还是降低可读性.....	427	816	REINTERPRET_CAST 运算符.....	455
第二十章 通用函数及模板.....		428	817	STATIC_CAST 运算符.....	456
780	模板.....	428	818	名字空间.....	456
781	使用简单模板.....	428	819	使用名字空间.....	457
782	支持多种类型的模板.....	429	820	对名字空间用 Using 语句	457
783	显式重载通用函数.....	430	821	运行时类型标识.....	457
784	通用函数的限制.....	430	822	为运行时类型标识使用 TYPEID.....	457
785	使用通用函数.....	431	823	TYPE_INFO 类	459
786	通用冒泡排序函数.....	432	824	MUTABLE 关键字	460