



面向 21 世纪 课 程 教 材
Textbook Series for 21st Century

高级语言 C++ 程序设计 编程范例与精解

周玉龙 刘 璟



高等教育出版社
HIGHER EDUCATION PRESS

内容提要

本书是《高级语言 C++ 程序设计》一书的配套教材。是专门针对《高级语言 C++ 程序设计》一书的各章节内容所精选而配制的 213 个编程范例。这些具有一定广度与深度的精选范例以及对它们的求解方法与过程, 非常适合刚刚开始学习 C++ 程序设计的学生、科技工作者以及自学者来参照模仿, 具有很高的实用参考价值, 可启发读者在程序设计方面的思维积极性, 提高面向对象程序设计方面的实用编程技术与技巧。本书的范例程序源代码可以在高等教育出版社网站上下载。

图书在版编目 (C I P) 数据

高级语言 C++ 程序设计编程范例与精解/周玉龙, 刘璟编. —北京: 高等教育出版社, 2003.4
ISBN 7-04-012300-2

I. 高... II. ①周... ②刘... III. C 语言-程序设计-高等学校-教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 009897 号

出版发行 高等教育出版社
社 址 北京市东城区沙滩后街 55 号
邮政编码 100009
传 真 010-64014048

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 北京中科印刷有限公司

开 本 787×1092 1/16
印 张 36
字 数 830 000

版 次 2003 年 4 月第 1 版
印 次 2003 年 4 月第 1 次印刷
定 价 39.00 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

计算机编程技术一直受到国内外各高等学府以及计算机界的高度重视。以面向对象技术为基础的 C++ 程序设计方法与技术代表了目前世界上最先进的计算机编程方法与技术。

学过和正在学习 C++ 程序设计方法与技术的人们可能都会有这样的感受,除了掌握语言的特定语法与使用规则外,学习与研究程序范例的作用也非常重要。通过学习、研究、模仿、进而想方设法改进与扩充程序范例的功能这一过程,也正是自己加深理解、学会使用并进一步提高编程能力的过程。为了对 C++ 语言进行有效使用尽快提高 C++ 程序设计方面的编程技术,能够顺利编制出各种具有较高水平的应用程序,学习研究并参照模仿一套 C++ 程序设计编程范例中所蕴涵的程序设计技术、技巧与经验,无疑是一条迅速提高自己编程技术与水平的捷径。基于上述想法,笔者编写了这本《高级语言 C++ 程序设计编程范例与精解》,以奉献给学习计算机编程技术的广大高校师生、科技人员以及计算机爱好者。

本书是《高级语言 C++ 程序设计》一书的配套教材。是专门针对《高级语言 C++ 程序设计》一书的各章节内容所精选而配制的 213 个编程范例。这些范例隶属于如下的十一个单元:第一章 C++ 语言初步;第二章 基本数据类型与基本运算;第三章 分支与循环处理;第四章 数组及其应用;第五章 函数的设计与应用;第六章 指针、结构体及引用;第七章 类与对象;第八章 继承与派生;第九章 运算符重载;第十章 模板及其应用;第十一章 流类库与输入/输出。

这是一套完整而又具有一定广度与深度的精选编程范例,这些范例以及对它们的求解方法与过程,非常适合于刚刚开始学习 C++ 程序设计的学生、科技工作者以及自学者来参照模仿,具有很高的实用参考价值,可启发读者在程序设计方面的思维积极性,提高面向对象程序设计方面的实用编程技术与技巧。例如,本书具有如下几方面的特色及其相关内容:

(1) 精选了一批具有一定深度的以及通过自定义类(类型)来实现面向对象编程的应用范例,既用于提高编程能力,又为今后开发面向对象的实用性软件奠定了基础。例如,八皇后问题,骑士巡游问题,万年历问题, n 元线性方程组求解,矩阵求逆等具有一定深度的问题;以及一批自定义的类(类型)及其对它们的具体使用——自定义的时间类、日期类、日期时间类、分数类、复数类、矩阵类、栈类、队列类、链表类、计算器类、学生类,等等。这些众多的自定义类中所涉及到的面向对象编程方法与技术是学习 C++ 编程技术的重点与难点,通过较多的编程范例来学习来模仿,进而掌握这些方法与技术是很有必要的,而这些内容在教科书或其他同类书中则很少见或只能见到其中的某一些。

(2) 对待同一个问题,要设法采用尽可能多的求解处理方法与实现技术,并对各种不同实现方法与技术的特点进行优劣性评价与比较。

学习研究 C++ 语言,灵活掌握与使用 C++ 语言为用户提供的各种数据类型以及用户自定义数据类型,并熟悉对各种不同数据的存储与处理方法和手段,是学好、用好 C++ 语言

的关键所在。为启发学生把学习重点放在编程技术与技巧方面，往往要求对同一问题尽可能采用多种不同解法，这样既可以提高对程序设计语言应用能力及其使用灵活性的理解，同时又能启发学生的思维积极性与创造性。例如对下述的同一个简单问题，本书就设计并给出了十种不同的求解方法（对应于如下编号的范例：1.5、4.1、4.14、5.23、6.7、6.17、6.18、7.4、11.17、11.18）。所要求解的问题为：从键盘输入 10 个 int 型数，而后反序输出它们。

十种实现方法的本质区别在于存放输入数据的结构与位置的不同，从而导致不同的程序处理方式。存放数据的位置与处理方式依次为：1. 使用 10 个 int 型变量；2. 使用 int 型数组；3. 使用 char 型字符数组(即字符串)；4. 使用递归函数中的局部变量；5. 使用通过运算符 new 动态申请到的数据空间；6. 使用指针与链表的栈式存储方式；7. 使用指针与链表的队列式存储方式；8. 使用自定义类中的私有数据成员；9. 使用 text 文件；10. 使用 binary 文件。实际上，213 个范例中还包含有许多一题二解、一题三四解的情况。

(3) 通过分析与探讨多种不同类型的具体实例，来提高对递归函数以及对问题的递归求解方法的理解，从而提高使用技术水平。例如，本书对 15 个不同问题采用了递归求解的方法，形成了 15 个不同的精选范例(5.19~5.25、5.30~5.34、5.40、5.47、5.48)。

递归函数以及对问题的递归求解方法是学习程序设计语言时的一个难点，本书着重从如何进行问题分解入手，通过分析与探讨多种不同类型的具体实例，来学习这种较难掌握的递归求解问题的方法与技术。即是说，主要是想办法将欲处理的“度为 n 的问题”分解化简为一个或若干个相同性质的“度小于等于 n-1 的问题”以及某些“本原问题”(不必再进行递归处理的“已解问题”)。另外程序员还应注意并保障递归函数在执行若干次后能够退出递归(不再进行递归调用，也即能够实现递归出口)。

(4) 含有若干个与绘图有关的范例及其相关的实现程序。在不同的操作系统与编译实现平台下，涉及到的绘图程序的编制方法可能有所不同。本书所有程序都是在 VC6.0 环境下实现的，绘图程序也不例外。实际上，在 VC6.0 环境下，可首先借助于 AppWizard 来自动生成一个典型的“Hello World”程序框架，而后在所生成的程序框架基础上，增加相关的“绘图”处理类以及处理 WM_PAINT 消息的“重画”处理函数等，进而实现所需的与绘图有关的程序功能。

需要注意的是，执行所生成的框架程序后进入的是 Windows 窗口模式(与 Dos 窗口模式不同，该模式支持鼠标操作，是以消息进行驱动执行的)，而且框架程序还自动为用户提供了可用的“File”和“Help”菜单以及一些简单的菜单命令项以及可用于进行窗口极大化、极小化、和关闭的鼠标按钮。这正是 VC6.0 之 AppWizard 的诱人之处，它为开发 Windows 程序提供了方便，使程序员不必再去为每一应用编写那些基本相同的框架式底层支撑程序(也即上述自动生成的程序框架)，而把注意力放在进行逻辑处理的所谓上层程序部分的编写上，最后再将它们衔接交融成同一个实现绘图功能的程序。

另外，对每一个范例都进行了精解，这意味着，并不是一步给出结果程序，而是把重点放在解释说明如何构造出这一程序，比如要重点解释说明程序编制中使用的数据结构、变量和语句的使用含义与相互关联性、实现方式以及此种实现方式的特点与优缺点评价等方面。实际上，许多看来相互无关的变量、语句、数据结构及实现方式，其实都是相互有关联的，只有深刻理解与体会到这种相互关联性，并能对它们进行合理与灵活的使用，才

可将程序设计技术与水平提升到一个新的高度。

书中所有范例所对应的程序都在 VC6.0 集成环境下调试通过，并附有可供参考的程序执行结果。

本书内容的选取与定夺是作者多年教学与科研中使用 C++ 经验的总结积累与结晶。我们期待并相信，本书的编程范例中所蕴涵的程序设计技术、技巧与经验必定会对广大读者起到较大的帮助与启发作用。

书中可能存在不少的错误与不足，敬请阅读本书的老师和同学们予以批评指正。

编者

2002 年于南开大学

目 录

第一章 C++语言初步	1
1.1 运行最简单程序	3
1.2 在屏幕上显示 3 行字符	3
1.3 对两整数进行四则运算	4
1.4 计算圆的面积与周长	5
1.5 输入 10 个整数而后反序输出的实现方式一	6
第二章 基本数据类型与基本运算	8
2.1 算术运算符功能的理解与使用	10
2.2 关系运算符和逻辑运算符功能的理解与使用	12
2.3 位运算符功能的理解与使用	13
2.4 表达式加括号使运算结果不变	15
2.5 使用自定义枚举类型	18
2.6 求两个时间(时刻)的时间差	19
2.7 计算 x 的平方、平方根及倒数等	21
2.8 一元二次方程求根	21
第三章 分支与循环处理	23
3.1 求三个实数中的最大数与最小数	24
3.2 设计一个计算商品价格的程序	25
3.3 switch 语句的执行效果	27
3.4 将考试成绩等级字母变换为相应的百分制分数段	28
3.5 求实数 a 的 n 次方	30
3.6 求出正整数 n 的所有因子	30
3.7 求正整数 n 的所有因子的平方根之和	31
3.8 n 个实数的整数部分和以及小数部分和	32
3.9 求最小公倍数	33
3.10 求最大公约数	34
3.11 改写程序使其中不出现 switch 与 for 语句	35
3.12 反序输出某正整数的各数位值	37
3.13 求正整数 n 之反序数的两倍值	38
3.14 验证奇数的指定特性	39
3.15 按近似公式求近似圆周率	40
3.16 三种控制语句的使用	41

3.17	求三种不同序列的累加和	43
3.18	求满足 eps 精度的累加和	45
3.19	处理从键盘输入的一批字符	46
3.20	将读入字符“拼装成”实数并显示出来	48
3.21	求 1 000 以内的所有“完全数”	52
3.22	求 100~999 内的所有“回文数”	53
3.23	编循环程序显示指定图形	54
3.24	理解二重循环程序的执行过程	55
3.25	根据要求显示具有不同特征的图形	56
3.26	循环累加不同的项数求出多个不同近似程度的 E 值	58
3.27	求满足给定两条件的最小自然数 n	59
第四章	数组及其应用	61
4.1	输入 10 个整数而后反序输出的实现方式二	62
4.2	二进制数转化为十进制数	63
4.3	十进制正整数转化为二进制	64
4.4	求字符串长度	65
4.5	两个字符串连接	66
4.6	字符串比较	67
4.7	求 1 000 以内的所有“完全数”并将结果按指定格式输出	69
4.8	求数组中的最大数和最小数	71
4.9	三种控制语句与数组的配合使用	72
4.10	对数组中的 10 个元素进行排序	74
4.11	合并两个有序数组	75
4.12	罗马数据转化为十进制数据	76
4.13	十进制正整数转化为罗马数据	78
4.14	使用最少数人民币纸币方案的求取	81
4.15	输入 10 个整数而后反序输出的实现方式三	83
4.16	计算某一天是星期几(方式一).....	85
4.17	屏幕显示某年某指定月的月历(方式一).....	86
4.18	求取正整数“圈”中连续四数之和最大者	89
4.19	将正整数转换为“英文表示字符串”	90
4.20	二维整型数组按行按列求出各累加和	92
4.21	二维数组的简单应用	93
4.22	二矩阵相加	95
4.23	处理二维字符数组	97
4.24	统计各行“单词”个数的处理方案一	99
4.25	统计各行“单词”个数的处理方案二	101
4.26	二矩阵相乘	103
4.27	在屏幕上“画出”最近 15 天的气温值柱形图	105
4.28	统计每一正整数的出现频率	107

第五章 函数的设计与应用	110
5.1 自定义函数及其调用	112
5.2 一般自定义函数及其调用	113
5.3 设计无参函数实现固定显示功能	114
5.4 设计一参函数使显示行数可变化	115
5.5 设计二参函数使变化更灵活	116
5.6 全局变量与无参函数配合使用	117
5.7 将十进制数转化为二进制数的自定义函数	118
5.8 将十进制数转化为八进制数的自定义函数	119
5.9 返回布尔值的函数	121
5.10 判断整数 x 是否仅由奇数字组成的自定义函数	122
5.11 判断正整数 n 是否为“回文数”的自定义函数	123
5.12 判断整数 n 是否为素数的自定义函数	124
5.13 返回整数 x 某一指定位之数值的函数	126
5.14 三种控制语句与数组及函数配合使用	127
5.15 计算两个日期之间的连续天数	130
5.16 计算出两个日期的间隔天数	133
5.17 计算某一天是星期几(方式二)	137
5.18 屏幕显示某年某指定月的月历(方式二)	139
5.19 使用三种方法求 Fibonacci 数列的某个项值	141
5.20 函数的递归调用	143
5.21 使用递归解法求实数 a 的 n 次方	145
5.22 使用递归方法打印一批数据	146
5.23 输入 10 个整数而后反序输出的实现方式四	147
5.24 按递归方法求最大公约数	148
5.25 读非递归函数后添加具有相同功能的递归函数	149
5.26 求字符串长度的函数	151
5.27 二进制数转化为十进制数的函数	152
5.28 查看数组中满足某条件之数据的个数	153
5.29 在升序数组中折半查找某元素序号的函数	154
5.30 显示数组某下标范围内所有元素的递归函数	156
5.31 判断两数组前 n 个元素值是否完全相同的递归与非递归函数	157
5.32 判断某数组前 n 个元素是否完全有序的递归与非递归函数	159
5.33 判断 s 字符串是否为“回文”的递归函数	160
5.34 函数的间接递归	162
5.35 两个二进制正整数的四则运算	164
5.36 两个罗马数据的四则运算	168
5.37 将 ASCII 码数字字符串转化为整数的函数	172
5.38 将整数转化为 ASCII 码数字字符串的函数	174
5.39 实现插入排序的 InsertionSort 函数	176
5.40 分析递归函数的执行结果	177

5.41	对两个字符数组实现某些统计功能的函数	179
5.42	不同作用域中同名标识符的处理	181
5.43	文件级作用域以及函数级作用域的变量	183
5.44	全局变量以及具有 static 存储类别变量的使用	185
5.45	extern 型外部变量的使用	186
5.46	与全局变量以及静态局部变量相关的使用	189
5.47	求解骑士巡游问题——为骑士求解巡游“路线图”	190
5.48	求解八皇后问题	195
第六章	指针、结构体及引用	199
6.1	与字符指针相关的使用	203
6.2	指针及其相关运算	204
6.3	判断字符串是否为合法标识符	205
6.4	在字符串数组中查找给定串 string 的出现位置	207
6.5	通过指针参数来实现“双向传值”	209
6.6	字符指针与其他类型指针的使用区别	211
6.7	输入 10 个整数而后反序输出的实现方式五	213
6.8	测试并运行含有 new 与 delete 的指定程序	214
6.9	实现具有动态大小的矩阵乘法	216
6.10	求解线性方程组	219
6.11	赋值参数、指针参数以及引用参数的使用区别	225
6.12	求满秩矩阵的逆矩阵	227
6.13	有关指针数组(各分量均为指针)的使用	232
6.14	指向一维数组的指针及其使用	234
6.15	结构体与数组两种数据结构的结合使用	236
6.16	添加函数完成与结构体有关的对复数的某些指定运算	239
6.17	输入十个整数而后反序输出的实现方式六	242
6.18	输入十个整数而后反序输出的实现方式七	244
6.19	使用由结构体形成的链表来实现数据排序	246
6.20	将某个有序链表合并(插入)到另一个有序链表之中	248
6.21	合并两个有序链表形成一个新的有序结果链表	252
6.22	利用链表存储字符串并实现相关的字符串处理	257
6.23	对函数指针的使用	261
6.24	使用返回指针值的函数	263
6.25	求取源串中首次出现某子串的头字符位置	264
6.26	通过带参数的 main 函数实现指定的四则运算	267
6.27	引用的建立及简单使用	269
第七章	类与对象	271
7.1	自定义类(类型)及其相关的构造与析构函数	274
7.2	动态对象的构造与析构	276

7.3 自定义的 Circle 类(类型)以及对它的使用	278
7.4 输入十个整数而后反序输出的实现方式八	280
7.5 自定义类及其有关类成员函数的使用	282
7.6 一个简单的复数类型 complex	285
7.7 学生类(类型)student	288
7.8 自定义的 queue (队列) 类型	291
7.9 在平面上用直线、矩形和圆画出一个粗略的汽车图形	295
7.10 含有静态数据成员及静态函数成员的自定义类	303
7.11 自定义类中含有常量数据成员	305
7.12 友元函数的使用	306
7.13 在复数类型 complex 中使用多个友元函数	309
7.14 简单的分数类(类型)fraction	312
7.15 “后缀式简单单步计算器”类(类型)Calculator	319
7.16 自定义时间类(类型)TimeType	329
7.17 在自定义时间类(类型)TimeType 中使用多个友元函数	336
7.18 自定义日期类(类型)DateType	343
7.19 自定义的计算器类(类型)compu	350
7.20 求满秩矩阵逆矩阵的自定义类 Matrix	360
第八章 继承与派生	366
8.1 继承与派生关系中相关成员的存取权限	368
8.2 学生教师档案的管理	372
8.3 派生类的构造函数及其对象的初始化	378
8.4 派生类的构造与析构函数及其对象的初始化	380
8.5 使用类定义方式求解线性方程组	383
8.6 通过组合方式自定义日期时间类(类型)DateTimeType	389
8.7 通过继承方式自定义日期时间类(类型)DateTimeType	394
8.8 虚函数的使用	398
8.9 通过虚函数手段按不同方法来计算定积分	401
8.10 增加梯形图元类后在平面上画出简单图形	407
8.11 派生关系中的赋值兼容性以及重名成员处理等	418
第九章 运算符重载	422
9.1 通过友元方式重载运算符	425
9.2 通过类成员方式重载运算符	428
9.3 自定义简单分数类(类型)fraction 中的运算符重载	431
9.4 在处理链表的 link 类中重载下标运算符	439
9.5 重载全局运算符 new 与 delete	442
9.6 对自定义类重载运算符 new 与 delete	444
9.7 全权控制运算符 new 与 delete 的分配与释放细节	447
9.8 自定义的类型转换运算符(函数)	450

9.9 成员指针运算符的使用	451
9.10 自定义矩阵 matrix 类(类型)中的运算符重载	453
第十章 模板及其应用	467
10.1 对一维数组进行排序的函数模板	469
10.2 二维数组中找最大元和最小元的函数模板	471
10.3 类模板的使用	474
10.4 既使用类型参数又使用普通参数的类模板	477
10.5 为链表类模板 list 补充成员函数	479
10.6 实现队列类模板 queue	484
10.7 实现数组类模板 array	488
10.8 使用了栈类模板的计算器类(类型) compu	493
10.9 类模板作基类派生出类模板	502
第十一章 流类库与输入/输出	506
11.1 插入与提取运算符的两种使用方式	509
11.2 对自定义 fraction 类重载输入运算符“>>”与输出运算符“<<”	511
11.3 使用格式控制符将数据输出到屏幕上	514
11.4 使用格式控制符将数据输出到 text 型磁盘文件中	515
11.5 从文件 cin 以抽取算符读入各字符的方式一	517
11.6 从文件 cin 以抽取算符读入各字符的方式二	519
11.7 从文件 cin 以 get 函数读入各字符	520
11.8 从磁盘文件 fin 以抽取算符读入各字符	521
11.9 使用 get 与 put 对指定文件进行拷贝	522
11.10 对磁盘文件的读/写处理(数值性数据)	523
11.11 对磁盘文件的读/写处理(文本性数据)	528
11.12 使用 getline 读出 text 文件中的“一篇文章”进行处理	531
11.13 使用 getline 读出 text 文件中的 C++ 源程序并删除注释	533
11.14 读出两个指定文件的各行内容进行比较	536
11.15 使用算符“>>”与“<<”对自定义的 text 型文件进行读/写	539
11.16 使用 write 与 read 对自定义的 binary 型文件进行读/写	542
11.17 输入 10 整数而后反序输出的实现方式九	545
11.18 输入 10 个整数而后反序输出的实现方式十	547
11.19 binary 型学生数据文件的简单管理	548
11.20 对 binary 型学生数据文件管理程序进一步扩充	552
参考文献	562

第一章 C++语言初步

[语法要点]

详细内容可参阅教材第二章。

(1) 每一个 C++ 程序都是一个基本符号的序列，基本符号包括大小写英文字母、数字(0~9)和特殊符号(33个)。

(2) 注释可以提高程序的可读性，在程序的执行中不起作用，在编译时被跳过。符号//表示该行中其右的所有内容为注释。符号“/*”和“*/”表示在其间的一行或多行内容为注释。

(3) C++ 程序中的基本语义单元是单词，由空格或分割符划分；若干单词按一定语法规则组成语句，例如，int a,b=3; 是一个说明语句，C++ 程序是一个语句序列。

C++ 的单词分为五类：**关键字**(key word)，**标识符**(identifier)，**字面常量**(literal constant)，**运算符**(operator)和**分割符**(separator)。

关键字 由系统指定，有固定含义，如 include、int、const、float 等。

标识符 是由用户指定的以字母或下横线开头的字母、数字、下横线序列，用来给变量、常量、函数、类、对象等语法元素起名；不可与关键字相同，区分字母的大小写(即 Abc、abc 是两个不同的标识符)，有一定的有效长度。

字面常量 包括十进制、八进制、十六进制整数、浮点数、字符常量(如“A”、“3”、“=”)和字符串常量(如“Hello world!”)。

运算符 一般由一个或多个特殊符号组成。如+、%、==、*=；也有些关键字如 sizeof、new、delete 也同时分列入运算符。

分割符 用来界定或分割其他语法成分。包括：空格、“、#、()、/*、*/、//、’、;、{} 等。

(4) **主函数 main** 是 C++ 程序中惟一必不可少的成分，最短的 C++ 程序为 main(){}，它是合法的，但什么也不做。C++ 程序中的主函数是整个程序的主控模块；又是程序的入口；还是程序与其运行环境的接口。

主函数名 main 是系统指定的标识符，主函数的返回类型为 void 或 int(可缺省)，一般主函数没有参数，其首部为 void main(void)。

(5) **预处理命令** 用来完成 C++ 程序编译前的“预处理”工作。预处理命令必须由“#”符号开头。预处理命令是从 C 语言中继承下来的，由于 C++ 语言本身和编程环境的发展，目前采用最多的是文件嵌入命令(# include)。

include 命令 用来把另一个称为头文件的内容在编译前嵌入到 # include 所在位置，与下面的程序一同作为源程序参加编译。

define 命令 称为宏定义命令，例如，# define PI 3.14，相当于把 PI 说明为常量 3.14，由于 C++ 中主张使用常量说明及内联函数，所以在 C++ 语言中 #define 命令的使

用比 C 语言中少得多。

if 和 **# endif** 等条件编译命令, 用来根据条件决定源程序中哪些参加编译, 哪些不参加编译。

(6) **输入/输出(I/O)** C++语言的标准 I/O 语句简明方便。

为了使用 **cout** 语句和 **cin** 语句进行输出输入, 要使用文件嵌入命令 “# include < iostream.h >”, 用来把系统提供的标准 I/O 流类的说明包含到源程序中。

cout 是系统指定的标识符, 用来表示输出的目标, 这里指屏幕, “<<” 称为插入运算符。

cin 则表示输入的来源, 这里指键盘, “>>” 称为提取运算符。

endl 标准标识符, 表示“回车”, 它与字符 “\n” 起相同作用, **cout << endl; cout << “\n”**, 都使打印头或光标移到下一行的左端。

C++语言保留了 C 语言中用于 I/O 操作的标准函数库, 其说明在头文件 **stdio.h** 中, 与上述 **cout <<**、**cin >>** 对应的 I/O 函数为 **printf()** 和 **scanf()**, 读者可参见教材附录 B。

(7) C++程序的运行

从编辑到运行 过程为编辑—预处理—编译—链接—运行。

在实际操作过程中, 编辑是由人工在计算机上完成的, 其余几个步骤则可由计算机完成, 不过如果出现故障, 调试的过程当然应由人工在计算机上完成。

Visual C++的集成环境 可以支持采用 C++语言进行的各种软件开发, 把开发过程按工程项目来管理, 因此其功能齐全, 系统复杂。与本书相关的上机实习题目一般比较简单, 只需要把从编辑到运行一个程序的基本过程搞清楚就可以了, 下面是最基本的操作步骤:

① 在“程序”表中选择 Microsoft Visual C++ 6.0, 进入到 Visual C++ 6.0 的集成环境窗口;

② 在“File”菜单下选择“new”项, 出现“new”对话框;

③ 在“new”对话框, 点击“Projects”标签, 在对话框中, 选择“win32 console Application”(win32 控制台应用); 在“Projects name”位置下输入项目名称(例如: demo); 在“Location”位置下, 选择工作区空间的位置, 就是确定新建的以 demo 为名的工作区目录的地址(例如: d:\mydir1), 完成后按“OK”按钮;

④ 在下一对话框中, 由用户选择项目类型, 例如, 可选择“An empty project”, 完成后按“Finish”按钮, 并在下一“报告”窗口选择表示接受的“OK”按钮;

⑤ 进入编程环境窗口, 窗口分为四部分。上部: 菜单和工具条; 中左: 工作区(workspace)显示窗口, 这里显示工作中与项目相关的各种文件种类等信息; 中右: 视图区, 是显示和编辑程序文件的操作区; 下部: 输出区, 程序调试过程中进行编译、链接、运行时输出的相关信息在这里显示。

⑥ 选择“Project/Add To Project/new”, 出现对话框, 点击“Files”并在出现的文件类型表中选择“C++ Source File”, 确认使右上角的项目名文本框中为“demo”, 其下的“File”文本框中添入一个源文件名, 如“main”, 并确认使下面的“Location”文本框中为工作区目录的位置路径“d:\mydir1”, 按“OK”回到主环境窗口。

⑦ 在屏幕中右的视图区(也即显示和编辑程序文件的操作区)中输入源程序。

⑧ 在“Build”菜单下，选“compile main.cpp”，进行编译；查看屏幕下部的输出区即“output”区的输出信息，如有出错信息，修改源程序重新编译。

⑨ 编译通过后，选“Build”菜单的“Build demo.exe”进行链接，如有出错信息，进行修改。

⑩ 选“Build”菜单的“Execute demo.exe”运行程序。

上述给出的是最基本的操作步骤，实际上，也可通过某些其他的选择或操作步骤来实现运行一个程序的目的，随着对 VC6.0 集成环境的逐步熟悉，将会一步步提高对它的使用技术与应用技巧。

1.1 运行最简单程序

在计算机上对教材 2.1.1 节的 program2_1.cpp 进行编辑、编译、连接、运行。

[分析提示]

参照教材附录 A，对程序进行编辑、编译、连接、运行。

[解答程序]

```
//program2_1.cpp
#include <iostream.h>
void main(void)
{
    cout<<"Let's learn to write a C++ program.";
}
```

[输出结果]

Let's learn to write a C++ program.

1.2 在屏幕上显示 3 行字符

编程序，执行后在屏幕上显示出如下的 3 行字符：

Hello, everyone!

How are you!

Good-by!

[分析提示]

通过使用如下形式的语句来显示 1 行字符：`cout<<".....\n";`其中的省略号代表要显示的字符串本身，最后的“\n”为转义字符，表示换行。

[解答程序]

```

#include <iostream.h>

//要往标准流对象 cout(即屏幕)上输出信息, 必须包含头文件
// "iostream.h"
//以双斜杠( "//" )开头直到行末的全体(符号)称为注释。
//在编译时注释将被当做空格处理

void main()
//任何一个 C++ 程序都含有一个必须叫做 main 的主函数
{
    cout<<"Hello, everyone!\n";    //显示第一行, 靠最后的 "\n" 实现换行
    cout<<"How are you!"<<"\n";    //显示第二行
    cout<<"Good-bye!"<<endl;    //显示第三行, endl 的功能与 "\n" 相同,
    //实现换行
}

```

[输出结果]

```

Hello, everyone!
How are you!
Good-bye!

```

1.3 对两整数进行四则运算

编程序, 从键盘输入两个 int 型(整型)数据, 而后通过使用运算符+、-、*、/、%来计算这两个数的和、差、积、商以及余数, 并将结果显示在屏幕上。

[分析提示]

(1) 要从键盘为变量输入数据, 可使用如下形式的语句来完成:

```
cin >> <变量> >> ... >> <变量>;
```

(2) cout 的输出项可以是表达式, 此时由 cout 负责将计算出的表达式结果值显示出来。如 "cout<<a+b;"。

[解答程序]

```

#include <iostream.h>

void main(void)
{
    int a,b;    //说明 a、b 为两个 int 型(整型)变量
    cout<<"Input 2 integers:"; //屏幕提示进行输入
    cin>>a>>b;    //从键盘输入两个数赋值给变量 a 与 b(两数据以空格分割)
    cout<<a<<"+"<<b<<"="<<a+b<<endl;
}

```

```

//在屏幕上显示出两数的和(和由计算机负责算出)
cout<<a<<"-"<<b<<"="<<a-b<<endl; //在屏幕上显示出两数的差
cout<<a<<"*"<<b<<"="<<a*b<<endl; //两数的积
cout<<a<<"/"<<b<<"="<<a/b<<endl; //两数的商
cout<<a<<"%"<<b<<"="<<a%b<<endl; //两数的余数
}

```

[输出结果]

Input 2 integers:11 3

11+3=14

11-3=8

11*3=33

11/3=3

11%3=2

1.4 计算圆的面积与周长

编程序实现：用户从键盘输入圆的半径(任一个浮点数 *radius*)，由程序负责计算出其相应的圆的面积与周长，并将计算结果显示出来。

[分析提示]

(1) 说明 *float* 类型的变量 *radius*、*area* 以及 *circum*，分别用来表示圆的半径、面积与周长；并说明一个 *float* 型的有名常量 *pai*，且将它的值设定为字面常量 3.14。

(2) 通过 *cin* 输入半径的值，而后通过 “*area=pai*radius*radius;*” 以及 “*circum=2*pai*radius;*” 来计算出相应的圆的面积与周长。

[解答程序]

```

#include <iostream.h>
/*
若要往标准流对象 cout(即屏幕)上输出信息，或者要从标准流对象 cin(即键盘)上输入
信息，必须包含头文件 “iostream.h”
*/
//以有序双符号 “/*” 开始(往后)，直到第一次遇见有序双符号 “*/” 为止的全体(符号)
//也称为注释。此种方式给出的注释可以占一行或多行
void main()
{
const float pai=3.14; //说明一个 float 型(浮点型)有名常量 pai
//并将它的值设定为字面常量 3.14，该有名常量 pai 的值在程序中保持不变

```

```

float radius; //说明一个 float 型(浮点型)变量 radius, 用于表示圆的半径
//并说明一个 float 型(浮点型)变量 circum, 用于表示圆的周长
cout<<"Enter radius:";
//在屏幕上显示 "Enter radius:" 符号串, 提示用户输入半径值
cin>>radius;
//等待用户从键盘输入一个浮点数, 并将其赋值给变量 radius
float area=pai*radius*radius;
//说明一个浮点型变量 area, 并将三个变量的乘积赋值给它
cout<<"\nThe area of circle is "; //在屏幕上先换行( "\n" 的功能)
//而后显示出: "The area of circle is" 符号串
cout<<area<<endl;
//在屏幕上紧接着显示 area 变量的值并换行
float circum=2*pai*radius;
//说明一个浮点型变量 circum, 并将三个变量的乘积赋值给它
cout<<"The circum of circle is "<<circum<<endl;
//在屏幕上先显示出: "The circum of circle is" 符号串
//而后紧接着显示 circum 变量的值并换行
}

```

[输出结果]

```

Enter radius:2.5
The area of circle is 19.625
The circum of circle is 15.7

```

1.5 输入十个整数而后反序输出的实现方式一

读下述程序, 并上机进行验证, 看是否实现了输入十个整数而后反序输出的预期目标。注意: 本程序使用十个 int 型变量存放输入数据。

```

#include <iostream.h>
void main(void)
{
    int a1,a2,a3,a4,a5,a6,a7,a8,a9,a10; //注 1
    cout<<"Input 10 integers:"<<endl; //注 2
    cin>>a1>>a2>>a3>>a4>>a5>>a6>>a7>>a8>>a9>>a10; //注 3
    cout<<"---- The result ----"<<endl;
    cout<<a10<<" "<<a9<<" "<<a8<<" "<<a7<<" "<<a6<<" "; //注 4
    cout<<a5<<" "<<a4<<" "<<a3<<" "<<a2<<" "<<a1<<endl;
}

```