



电脑技能百练丛书



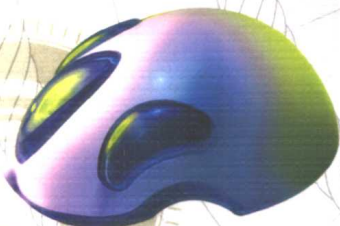
C++

程序设计

技能百

练

蒋立翔 编著



定价

35
元

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



C++程序设计 技能百练

蒋立翔 编著

中国铁道出版社

2004 • 北京

(京)新登字 063 号

内 容 简 介

C++是最完善的程序设计语言之一,是高级程序员的首选。要想成为一个真正的程序员,必须熟练掌握 C++语言。

本书融趣味性、实用性于一体,通过 100 个经典实例,全面讲解 C++程序设计的方法和步骤。第 1 课讲解 C++基础知识;第 2 课讲解数据类型和控制结构;第 3 课讲解函数;第 4 课讲解数组和指针;第 5 课讲解引用与结构;第 6 课讲解类和对象;第 7 课讲解堆、构造函数与拷贝构造函数;第 8 课讲解静态成员、友元与模板;第 9 课讲解继承与派生;第 10 课讲解多态性和虚函数;第 11 课讲解 I/O 流类库和异常处理。

本书资料丰富、内容详实,涵盖了 C++编程的难点和热点。既适于作为大中专计算机及相关专业及培训班的 C++教程,也可以作为广大编程爱好者的自学教材。

图书在版编目(CIP)数据

C++程序设计技能百练/蒋立翔编著. —北京:中国铁道出版社,2003.10

(电脑技能百练丛书)

ISBN 7-113-05557-5

I. C… II. 蒋… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 092756 号

书 名: C++程序设计技能百练

作 者: 蒋立翔

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

责任编辑: 严晓舟 王占清

封面设计: 孙天昭

印 刷: 北京市彩桥印刷厂

开 本: 787×1092 1/16 印张: 21.75 字数: 520 千

版 本: 2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05557-5/TP·1054

定 价: 35.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

出版寄语

技能百练含意

俗话说：千锤百炼，百炼成钢。不经历风雨，怎能见彩虹？

学电脑的路上，有你，有我，并不孤单。但你我用同样的鼠标，同样的键盘，同样的电脑配置，做出来的效果，却总不一样，这是为什么？

这是因为，一分耕耘，一分收获。

当然，不止这些，因为没有创意，作品就没有灵魂。

所以，聪明应该转变成智慧，智慧激发灵感。

在电脑这个广阔的天地里，我们从来不停步。夜深了，击键声伴着家人的鼾声，时间从光标的箭头下飞逝。我们还不能歇息，电脑时代赋予我们去创造具有生命力的作品。

但这一切，都有赖于我们掌握了非常扎实的基础知识和技能。事实上也的确如此，君不见古埃及大小金字塔，因其底座大小不同，高低亦不同，底座越大，金字塔越高。

有鉴于此，我们推出《电脑技能百练丛书》，旨在伴您在电脑的旅途上一路同行！

丛书编委

本丛书由第一时间工作室创作完成。愿我们在第一时间共同感受来自电脑时代的最强音。

本丛书编委会成员如下：陈艳红、谭雨、翟杰、丛丽丽、岳辉、亚敏德、马小乐、任飞、莫非、袁玉民、薛卫红、顾成成、蒋立翔、季子云、郑星、张永可、李沙、宋妍、姬晓林、杨阳、林慕新、寇婉如、刘丽丽、万雪莲等。

前言



比尔·盖茨有句名言：“Just thinking of things as winning is a terrible approach. Success comes from focusing in on what you really like and are good at—not challenging every random thing”（把一切都看作成功是一种可怕的态度，成功来自于你對自己真正热爱和擅长的事业的专注——而不是来自于对每一件偶然事情的挑战）。比尔·盖茨在战略的高度回答了怎样才能成功这一重大问题。

学习 C++ 语言何尝不是这样！C++ 是功能最强大、最完善的计算机语言之一，许多人都想学好它，但它却令初学者望而却步、无所适从。所以，要学好 C++ 语言，就要培养自己对它的真正热爱，然后专注于它。

编写本书的目的就是给有志于学习 C++ 或 VC（Microsoft Visual C++ 的简称）的读者引路、排忧解难，让大家轻松自如地迈进 C++ 的神秘殿堂，领略 C++ 的独有风姿。

本书是作者几年学习与应用 C++ 经验与体会的总结，里面包含了作者几年前初学 C++ 时的诸多困惑（如指针及其相关应用、算法设计的程序实现、面向对象概念的建立等），有时甚至是学习中的迷惘。也因此，本书在讲解中时刻不忘答疑解惑。

本书有以下特点：（1）采用实例教学法，将每一个知识点都融进 100 个实例之中，易学易懂。（2）知识点突出，每一个实例将用到的知识点单独以表格形式列于实例之前，并将新学知识点与已学知识点分开，便于加深理解与记忆。（3）所举实例力求趣味性与实用性的统一。如“实例 2 分牛的传说”、“实例 10 验证‘歌德巴赫猜想’”、“实例 11 魔方阵”、“实例 13 八皇后游戏”、“实例 22 智者生存”等实例具有很强的趣味性。“实例 14 探询路径”、“实例 23 矩阵转置”、“实例 32 编排日历”、“实例 47 Huffman 编码”、“实例 97 我的朋友信息管理”等实例具有很强的实用性。（4）特别注重各门相关课程的衔接与融合。本书所举实例几乎涵盖了数据结构课程的所有重要内容（如：链表及其应用、队列及其应用、栈及其应用、二叉树及其应用、图及其应用、各种排序算法、文件及其管理）；有大量算法设计及其分析课程的内容（几乎所有的重要算法本书都有体现，如：递归法、动态规划法、回溯法、概率算法、近似算法等）；有体现人工智能方面知识的内容（如：“实例 17 比我疆土”、“实例 18 巧填数字”、“实例 49 农夫、狼、羊与白菜问题”等）。（5）重难点知识清晰（如指针及其应用、面向对象概念的建立等），加重份量进行练习。

本书共分 11 课，第 1 课～第 5 课讲述 C 语言到 C++ 的相关知识和结构化程序设计方法（因为 C++ 是从 C 语言基础上发展起来的，没有扎实的 C 语言基础很难学好），同时渗透了大量的算法（如递归法、动态规划法、回溯法、概率算法、近似算法等）实践；第 6 课以后讲述 C++ 语言的相关知识和面向对象程序设计的方法。

本书所有实例均在 Visual C++6.0 环境下编译通过，所有实例源代码随光盘附送。

由于作者水平有限，书中错误和不当之处敬请专家和读者批评指正。并可直接与作者联系：jlx9524@sina.com.cn 或 wwwl@publicb.bta.net.cn。我们也会在适当时间进行修订和补充，并发布在天勤网站：<http://www.tqbooks.net> “图书修订”栏目中。

蒋立翔

2003 年 11 月

目 录

第 1 课 C++基础知识	1
课堂讲解	2
一、从 C 到 C++	2
二、程序与语言	2
三、结构化程序设计（为处理复杂问题提供了有力的手段）	3
四、面向对象程序设计	4
五、程序开发过程	4
六、C++程序基本框架结构	4
七、结构化程序设计与面向对象的程序设计比较	6
八、编译环境	6
九、Microsoft Visual C++的帮助	8
十、Visual C++开发环境概述	8
上机练习	10
练习 1 作者的祝愿	10
练习 2 分牛的传说	12
第 2 课 数据类型和控制结构	15
课堂讲解	16
一、字符集与保留字	16
二、数据类型	16
三、变量定义	17
四、常量	17
五、I/O 流控制	18
六、表达式	18
七、算术类型转换	18
八、增量与减量	19
九、关系运算	19
十、逻辑运算	19
十一、条件运算符	19
十二、逗号运算符	19
十三、if 语句	19
十四、while 语句	19
十五、do ... while 语句	20

十六、for 语句	20
十七、switch 语句	20
十八、转向语句	20
上机练习	21
练习 3 打印图案	21
练习 4 大写金额	24
练习 5 证明“素数公式”	28
练习 6 牛顿迭代法求方程根	30
练习 7 二分法求方程根	32
练习 8 杨辉三角的奥妙	34
练习 9 判断整除	36
第 3 课 函数	39
课堂讲解	40
一、函数概述	40
二、函数原型	40
三、全局变量与局部变量	41
四、函数调用机制	42
五、递归 (Recursive) 函数	42
六、内联函数	42
七、重载函数	43
八、默认参数的函数	43
上机练习	44
练习 10 验证“歌德巴赫猜想”	44
练习 11 魔方阵	46
练习 12 汉诺塔 (Tower of Hanoi) 的递归解法	48
练习 13 八皇后游戏	51
练习 14 探询路径	54
练习 15 “掷双骰”游戏	56
练习 16 “完全数”判别 (方法一)	58
练习 17 比我疆土	61
练习 18 巧填数字	63
练习 19 打印螺旋方阵	64
第 4 课 数组和指针	66
课堂讲解	67
一、数组定义	67
二、数组的使用	67
三、向函数传递数组	67
四、二维数组	67

五、指针的概念	68
六、指针运算	69
七、指针与数组	69
八、堆内存	69
九、const 指针	69
十、指针与函数	69
十一、字符指针	70
十二、指针数组	70
十三、函数指针	71
上机练习	72
练习 20 “完全数”判别(方法二)	72
练习 21 Eratosthenes 法求素数	73
练习 22 智者生存	75
练习 23 矩阵转置	77
练习 24 字符串倒序(方法一: 指针实现)	80
练习 25 姓氏排序	82
练习 26 矩阵的马鞍点	84
练习 27 动态分配数组的存储空间	86
练习 28 寻找最长的单词	88
练习 29 const 指针	90
练习 30 顺序查找	91
练习 31 折半查找	93
练习 32 编排日历	94
练习 33 表白爱之“心”	97
练习 34 求定积分(梯形法)	98
第 5 课 引用与结构	101
课堂讲解	102
一、引用的概念和操作	102
二、用引用传递函数参数	103
三、结构	103
四、链表结构	104
上机练习	106
练习 35 求学位课平均分	106
练习 36 统计优秀者人数	108
练习 37 最大子段和(解法 1)	109
练习 38 最大子段和(解法 2)	111
练习 39 最大子段和(解法 3)	113
练习 40 100 米比赛排名(解法 1)	116

练习 41	100 米比赛排名 (解法 2)	118
练习 42	建立我的好友档案	120
练习 43	完善我的好友档案 (一)	123
练习 44	完善我的好友档案 (二)	125
练习 45	公正的计票器	129
练习 46	汉诺塔 (Tower of Hanoi) 的非递归解法	132
练习 47	Huffman 编码——建立 Huffman 树	134
练习 48	Huffman 编码——Huffman 树编码	138
练习 49	农夫、狼、羊与白菜问题——图的存储	140
练习 50	农夫、狼、羊与白菜问题——图的遍历	143
第 6 课	类和对象	148
课堂讲解		149
一、类的定义		149
二、成员函数的定义		149
三、对象的定义及其成员的访问		151
四、静态成员		152
五、对象的作用域和生存期		152
六、类成员的访问控制说明符		152
七、屏蔽类的内部实现的好处		152
上机练习		153
练习 51	斐波那契数列的计算	153
练习 52	计算两点的距离	156
练习 53	计算圆柱体体积与表面积 (结构实现)	159
练习 54	计算圆柱体体积与表面积 (类实现)	161
练习 55	建立学生成绩管理系统	163
练习 56	字符串倒序 (方法二: 栈实现)	168
练习 57	判断字符串是否回文	170
练习 58	求学位课平均分 (嵌套类实现)	173
练习 59	计算 π 值 (随机投点法)	177
练习 60	模拟抛硬币所得正面的频率图	180
第 7 课	堆、构造函数与拷贝构造函数	182
课堂讲解		183
一、构造函数		183
二、析构函数		184
三、拷贝构造函数		184
四、类的聚集 (组合)		185
五、初始化表对常量数据成员或常量引用成员提供初值		185
六、动态存储		186

上机练习	188
练习 61 复数的运算(成员函数实现)	188
练习 62 100 米比赛排名(解法 3)	191
练习 63 设计我的计算器	194
练习 64 学生信息的管理	199
练习 65 this 指针的应用	202
练习 66 类成员初始化的困惑	204
练习 67 避免缺省参数构造函数的二义性	208
练习 68 常数据成员的初始化	211
练习 69 拷贝构造函数(一)	213
练习 70 拷贝构造函数(二)	217
第 8 课 静态成员、友元与模板	220
课堂讲解	221
一、静态成员	221
二、友元	222
三、模板	223
上机练习	226
练习 71 金卡账目管理	226
练习 72 复数的运算(友元函数实现)	229
练习 73 求两直线的交点	231
练习 74 快速排序算法	234
练习 75 二分查找算法	236
第 9 课 继承与派生	239
课堂讲解	240
一、基类和派生类	240
二、子类型	242
三、虚基类	242
上机练习	244
练习 76 长途电话计费程序	244
练习 77 一个小公司的工资管理系统	247
练习 78 学生成绩管理系统(类继承)	251
练习 79 大学人员管理系统	254
练习 80 子随父姓	258
第 10 课 多态性和虚函数	261
课堂讲解	262
一、函数重载	262
二、运算符重载	263
三、静态联编和动态联编	264

四、虚函数	264
五、纯虚函数和抽象类	265
六、虚析构造函数	266
上机练习	266
练习 81 重载实现有理数的各种运算	266
练习 82 点对象间的各种运算	270
练习 83 比较同学的年龄	273
练习 84 矩阵的基本运算	276
练习 85 检查数组是否越界	279
练习 86 计算三角形面积之和	281
练习 87 链表指针对集合的各项操作	283
练习 88 抽象类求图形面积和周长	288
练习 89 虚重载运算符函数	290
练习 90 一个小公司的工资管理系统（虚函数实现）	293
第 11 课 I/O 流类库和异常处理	298
课堂讲解	299
一、I/O 流的概念	299
二、输出流	299
三、输入流	301
四、I/O 运算符重载	302
五、异常处理	302
上机练习	304
练习 91 显示九九乘法表	304
练习 92 浏览文本文件	306
练习 93 文本文件复制	308
练习 94 有理数的四则运算（重载“<<”、“>>”）	311
练习 95 我的朋友信息管理（一）	315
练习 96 我的朋友信息管理（二）	318
练习 97 我的朋友信息管理（三）	322
练习 98 异常处理应用	327
练习 99 显示文件大小	330
练习 100 异常处理中的构造与析构	331

第 1 课 C++基础知识

▼ 本课学习的主要内容有哪些？

C++语言是一门适合编写系统描述和应用软件的程序设计语言，它是一种既面向对象又面向过程的混合型程序设计语言。C++是在 C 语言的基础上发展起来的。开发 C++程序常见的软件有 Visual C++、C++ Builder 和 Turbo C++等。由于 Visual C++现在最为流行，本书例题也都在 Visual C++环境下运行调试通过，故读者应该熟悉 Visual C++集成开发环境。了解 C++语言的背景知识、C++与 C 的联系与区别、程序和语言的基本知识、结构化程序设计与面向对象的程序设计比较、程序开发的简单过程以及 C++程序的基本组成与框架结构等知识。

▼ 通过本课的学习要达到什么目标和要求？

通过本课的学习要使学员了解 Visual C++的程序开发环境，熟悉 C++语言程序的编写、编译、链接及运行的工作过程，理解 C++与 C 的联系与区别，掌握 C++程序语言的基本结构，了解如何通过 Visual C++的应用程序向导开发简单的应用程序。

▼ 本课知识对掌握 C++有何重要性？

C++语言是目前世界上最流行和应用范围最广的一种计算机高级程序设计语言。C++对 C 语言不是简单的扩充，而是从面向过程的语言发展为既面向过程又面向对象的语言，从而适应软件业发展的客观需要。事实上大部分 C++程序员有一定 C 语言基础，在此建议没有 C 语言基础的读者先学习以下 C 语言。本章起统帅全书的作用，也是进一步学习的基础，希望读者认真学习，领会实质。

▼ 本课难以理解的知识有哪些？掌握这些难点知识有什么方法和窍门？

本课是使用 Visual C++编程的基础，要好好练习。同时 Visual C++的开发环境、C++与 C 的联系、C++程序的基本组成和框架结构是难点。掌握本课内容需要对 Visual C++的开发环境有深入的了解，学习时可结合上机练习加以巩固。要掌握本课内容，可参阅有关 Visual C++的有关书籍，读者也可以通过上机编程体会这些程序的基本组成和框架结构。然后再结合本课的各个练习，以达到融会贯通的目的。刚入门学习时，重点应掌握基本概念、基本技巧、基础知识，选一些简单的典型的有实用性及趣味性的习题加以练习。

技能
百练

课堂讲解

一、从 C 到 C++

1. C 的优点

- (1) 语言简洁、紧凑, 使用灵活方便 (只有 32 个关键字, 程序书写自由);
- (2) 运算符和数据类型丰富;
- (3) 可以直接访问内存地址, 能进行位操作 (使之能够胜任开发 OS——操作系统);
- (4) 生成的目标代码质量高, 程序运行效率高, 可移植性好。

2. C 的缺点

- (1) 类型检查机制相对较弱, 程序中的一些错误不能在编译时发现;
- (2) 本身几乎没有支持代码重用的语言结构。注意函数是一种“语句的集合”, 它是将可以重复使用的语句集合在一起, 并给定一个代号来作为函数名称, 函数的引入减少了程序编写过程中的重复性劳动, 但函数不是 C 语言特有的, 几乎所有语言都支持相应的函数, 这一点 C++ 语言则有很大发展, 如 C++ 的继承性、多态性等 (后面将陆续学习);
- (3) 程序规模大时, 程序员很难控制程序的复杂性。

3. C++ 的产生

- (1) 为解决 C 存在的缺陷, 1980 年贝尔实验室开始对 C 进行改进和扩充;
- (2) 最初的成果称为“带类的 C”;
- (3) 1983 年正式取名为“C++”;
- (4) 1994 年制定了 ANSI C++ 标准草案。

二、程序与语言

1. 程序 (定义众多)

对计算机而言, 程序就是一组机器指令;

对人而言, 程序就是用某种高级语言编写的语句序列;

程序是任何有目的的、预想好的动作序列, 即一种软件;

计算机要运转起来, 需要一整套可运行软件, 即计算机程序。

2. 程序语言的发展

最原始: 计算机指令, 即机器语言程序、一串二进制数。

随后: 汇编语言, 可以将机器指令映射为一些能够被人读懂的助记符 (如 ADD, SUB)。

现今: 高级语言。

3. 程序语言的目标

计算机程序的最初目标：代码更短小，运行速度更快（由于计算机硬件和运行成本的下降，目标也发生了改变）。

随后目标：在程序正确的前提下，可读性、易维护、可移植性成了首要目标。

4. 高级语言和低级语言

高级语言：C++，C，Fortran，BASIC 等。

低级语言：机器语言，汇编语言。

程序语言越低级，描述程序越复杂，指令越难懂，可维护性越差。

程序语言越高级，越接近人的表达与理解，描述程序越简单，指令越易懂，可维护性越好。

C++语言的设计者 Bjarne Stroustrup 博士（注：Bjarne Stroustrup 是 C++语言的设计者，美国贝尔实验室研究员，该实验室“大型编程研究”部门领导人，1993 ACM Grace Murray Hopper award 的得主。编著了多本阐述 C++程序设计与理念的书，如“The C++ Programming Language”、“The Design and Evolution of C++”、“The Annotated C++ Reference Manual (ARM)”，其中的“The C++ Programming Language”被誉为 C++设计员必读的 Bible（圣经））在一篇论文中指出，C++设计的主要目标包含下列几点：

- 一个“更好的 C 程序语言”。
- 支持抽象化数据形式。
- 可以实现面向对象程序设计。

三、结构化程序设计（为处理复杂问题提供了有力的手段）

1. 主要思想

对需要解决的软件问题一般采用由上而下、由大到小、将问题逐步分解成较小的问题，直到可以使用程序语言轻易就能解决为止，这就是模块化概念。模块化设计在结构化分析设计时，一般是以“函数”为单位，“函数”可以将数据作特定处理来完成特定的工作。也就是说传统的程序化语言（Procedural Language）的程序设计中，是将数据和处理数据的程序分离来考虑的。这样可能会带来如下的问题。

2. 带来的问题

- 如果一个问题经系统分析设计后，要由两个函数完成，程序员可能会根据他自己对某一个函数模块内的实际细节的了解程度，在另一个不同的模块中制作并留下他所需要的结果，于是，这两个模块之间便产生了相依性，系统会变得更加复杂，更加不易维护。
- 程序员可能需要撰写不同的模块来存取相同的数据结构。由于这些模块是通过数据来彼此结合的，这将增加出现错误的可能性，以及修改的困难度。当这些共享的数据的存放形式或格式需要修改或增删时，程序员必须把所有用到这些共享数

据的模块全部找出，并适度地修改其程序代码，这是一件耗时费事，又容易出错的事。

- 模块化观念下的结构化分析设计，是设计人员为了解决问题而定出的规则，所以较不接近真实世界的问题，设计出的程序交互性差。

四、面向对象程序设计

1. 本质

把数据和处理数据的过程当成一个整体——对象（其实现：需要封装和数据隐藏技术，需要继承和多态性技术）。

2. 封装和数据隐藏

封装性：自成一体性；

数据隐藏：使用其功能，无需知道单元内部的工作原理；

C++中，可以将一个定义好的类看成完全封装的实体，作为一个整体单元使用。

3. 继承和重用

C++支持重用的思想，程序可以在扩展现有类的基础上，声明新的类。

新子类是从现有类派生出来的，称为派生类。

4. 多态性

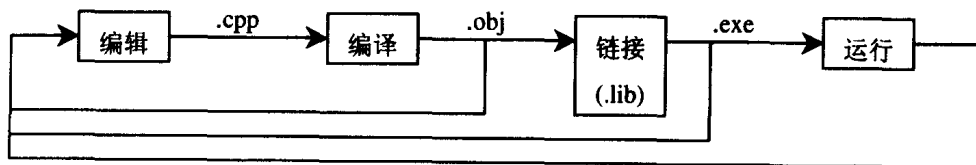
通过继承的方法构造类，采用多态性为每个类指定表现行为。

继承性与多态性的组合，可以生成一系列类似但独一无二的对象。

继承性使一些对象共享许多相似的特征。

多态性使每个对象具有独特的表现方式。

五、程序开发过程



六、C++程序基本框架结构

一般来说，C++基本框架结构包含声明区、主程序区和函数定义区3部分。

1. 声明区

声明区处在程序的所有函数的外部，它主要包含以下内容（不是每一程序都需要）：

- 文件：如`#include<iostream.h>`，`#include <string.h>`；

- 宏定义：如 `#define PI 3.14;`
- 类定义：如 `class student{.....};`
- 结构体定义：如 `struct animal{.....};`
- 联合体定义：如 `union area{.....};`
- 函数声明：如 `int add();`
- 全局变量（或符号常量）的声明：如 `float a =2.1;const double pi(3.14);`
- 条件编译：如 `#ifdef.....`。

2. 主程序区

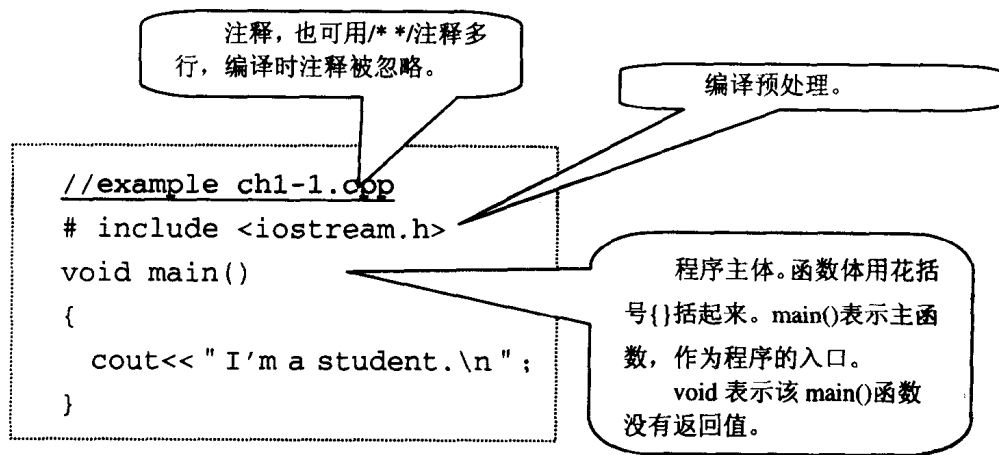
C++程序由函数构成，它总是从主函数 `main()` 开始运行，每个程序必须且只能有一个 `main()` 函数。该函数主要包含以下内容（不是每一程序都需要，视情况而定）：

- 局部变量声明：如 `int i =2;`
- 函数调用：如 `return add(.....);`
- 程序功能实现语句：如 `if.....;`
- 对象与结构处理等。

3. 函数定义区

C++是函数驱动的，函数定义是描述一个函数执行算法的过程。每个函数都是由函数说明和函数体两部分组成。

一个最简单的程序示例如下：



此程序结构包含：注释、编译预处理、程序主体。

(1) 注释：编译时，把每个注释都视为一个空格。

(2) 编译预处理：以 `#` 开头，“`# include <iostream.h>`”作用是将文件“`iostream.h`”的内容，在编译之前加入到文件“`ch1-1.cpp`”中（作为其中的一部分）。

另外，“`iostream.h`”是系统定义的一个头文件，它设置了 C++ 的 I/O 相关环境。

(3) 程序主体：函数定义：描述一个函数执行算法的过程。

七、结构化程序设计与面向对象的程序设计比较

(1) 结构化程序设计以过程为中心，数据和处理数据的过程分离，程序重用性差，维护困难。

(2) 面向对象程序设计中过程与数据结合在一起形成对象实体，对象具有继承性与封装性，代码重用性强。

(3) 面向对象程序设计继承了结构化程序设计的优点，并实现了更高层次抽象。

八、编译环境

目前常用来写 C++ 程序语言的工具有 VC++6.0、Borland C++Builder。本书所有实例在 Visual C++ 6.0 环境下编译通过。

本书所举实例在 VC++6.0 下编写步骤如下：

步骤 1：首先启动 VC++6.0，启动后画面如图 1-1 所示。

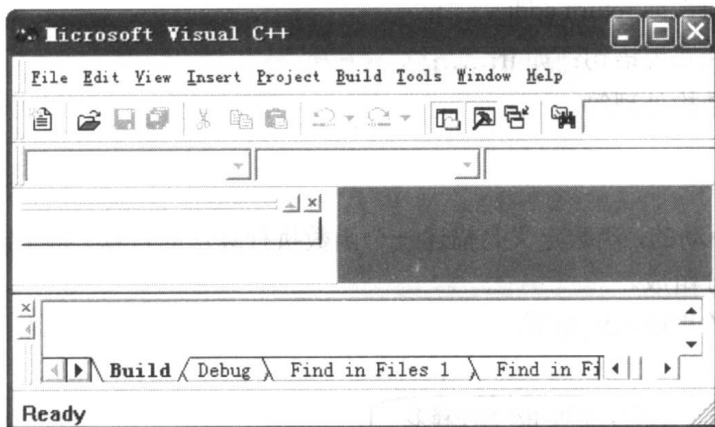


图 1-1

步骤 2：依次点选菜单上的 File/New 选项，用来建立新文件，如图 1-2 所示。

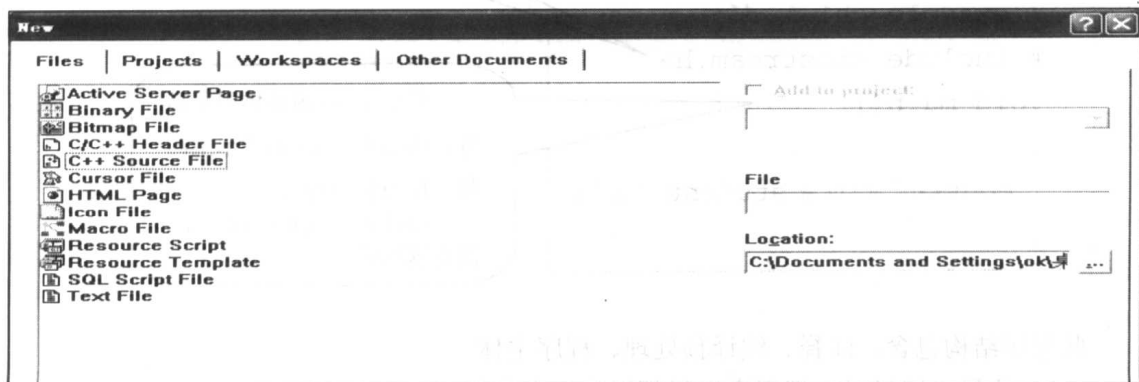


图 1-2

步骤 3：依次点选菜单上的 Files/C++ Source File 选项，输入文件名，在 Location 中选择其存放路径，如图 1-3 所示。