普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

高职高专教材系列

C++程序设计实用教程

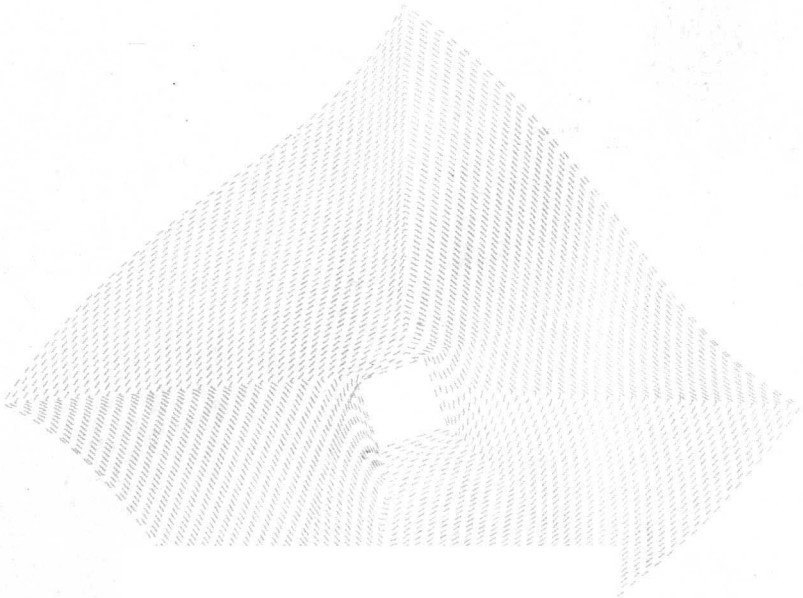
岳俊梅 李庆义 张峰 等 编著

清华大学出版社

丛书主编 谭浩强
高等院校计算机应用技术规划教材
高职高专教材系列

C++程序设计实用教程

岳俊梅 李庆义 张峰 等 编著



清华大学出版社
北京

内 容 简 介

本书内容详尽，示例丰富。通过本书的学习，读者可以了解C++语言及其程序的一般形式，并逐步掌握C++语言的核心内容，包括引用、重载、类和对象、继承和类的派生、多态性和虚函数、输入/输出流、调试初步等。本书主要针对面向对象程序设计的三大特征——封装性、继承性和多态性做了重点的描述。第8章可以使学生初步了解一些VC++调试工具的使用方法并能进行简单的调试工作，更重要的是可以利用逐步执行程序的调试方法来加深对各知识点的理解。

本书实用性和可操作性强，提供了大量编程示例及说明，可以帮助学生透彻理解所学的概念。通过大量的练习题进一步巩固所学的内容，确保学生能够真正掌握各章节的内容。

本书既可以作为高等院校计算机及相关专业的专业基础课教材，也可供各类软件开发人员参考。

图书在版编目(CIP)数据

C++程序设计实用教程/岳俊梅等编著. —北京：清华大学出版社，2007.1
高等院校计算机应用技术规划教材
ISBN 978-7-302-14114-3

Ⅰ. C…　Ⅱ. 岳…　Ⅲ. C语言—程序设计—高等学校—教材　Ⅳ. TP312

中国版本图书馆CIP数据核字(2006)第134543号

责任编辑：谢　琛　孟毅新
责任校对：李　梅
责任印制：何　芊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京季蜂印刷有限公司
装 订 者：三河市兴旺装订有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：15.5　**字　数**：354千字
版　　次：2007年1月第1版　　**印　次**：2007年1月第1次印刷
印　　数：1～4000
定　　价：22.00元

编辑委员会

《高等院校计算机应用技术规划教材》

《高等院校计算机应用技术规划教材》

进入21世纪，计算机成为人类常用的现代工具，每一个有文化的人都应当了解计算机，学会使用计算机来处理各种的事务。

学习计算机知识有两种不同的方法：一种是侧重理论知识的学习，从原理入手，注重理论和概念；另一种是侧重于应用的学习，从实际入手，注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对多数人来说，计算机是作为一种工具来使用的，应当以应用为目的、以应用为出发点。对于应用性人才来说，显然应当采用后一种学习方法，根据当前和今后的需要，选择学习的内容，围绕应用进行学习。

学习计算机应用知识，并不排斥学习必要的基础理论知识，要处理好这二者的关系。在学习过程中，有两种不同的学习模式：一种是金字塔模型，亦称为建筑模型，强调基础宽厚，先系统学习理论知识，打好基础以后再联系实际应用；另一种是生物模型，植物并不是先长好树根再长树干，长好树干才长树冠，而是树根、树干和树冠同步生长的。对计算机应用性人才教育来说，应该采用生物模型，随着应用的发展，不断学习和扩展有关的理论知识，而不是孤立地、无目的地学习理论知识。

传统的理论课程采用以下的三部曲：提出概念一解释概念一举例说明，这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者，我们提倡新的三部曲：提出问题一解决问题一归纳分析。传统的方法是：先理论后实际，先抽象后具体，先一般后个别。我们采用的方法是：从实际到理论，从具体到抽象，从个别到一般，从零散到系统。实践证明这种方法是行之有效的，减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

检查学习好坏的标准，不是“知道不知道”，而是“会用不会用”，学习的目的主要在于应用。因此希望读者一定要重视实践环节，多上机练习，千万不要满足于“上课能听懂、教材能看懂”。有些问题，别人讲半天也不明白，自己一上机就清楚了。教材中有些实践性比较强的内容，不一定在课堂上由老师讲授，而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力，启发

学生的求知欲望。

全国高等院校计算机基础教育研究会历来倡导计算机基础教育必须坚持面向应用的正确方向，要求构建以应用为中心的课程体系，大力推广新的教学三部曲，这是十分重要的指导思想，这些思想在《中国高等院校计算机基础课程2006》中作了充分的说明。本丛书完全符合并积极贯彻全国高等院校计算机基础教育研究会的指导思想。

这套《高等院校计算机应用技术规划教材》是根据广大应用型本科和高职高专院校的迫切需要而精心组织的，其中包括3个系列：

(1) 应用型教材系列。适用于培养应用性人才的本科院校和基础较好、要求较高的高职高专学校。

(2) 高职高专教材系列。面向广大高职高专院校。

(3) 实训教材系列。应用型本科院校和高职高专院校都可以选用这类实训教材。其特点是侧重实践环节，通过实践(而不是通过理论讲授)去获取知识，掌握应用。这是教学改革的一个重要方面。

本套教材是从1999年开始出版的，根据教学的需要和读者的意见，几年来多次修改完善，选题不断扩展，内容日益丰富，先后出版了60多种教材和参考书，范围包括计算机专业和非计算机专业的教材和参考书；必修课教材、选修课教材和自学参考的教材。不同专业可以从中选择所需要的部分。

为了保证教材的质量，我们遴选了有丰富教学经验的高校优秀教师分别作为本丛书各教材的作者，这些老师长期从事计算机的教学工作，对应用型的教学特点有较多的研究和实践经验。由于指导思想明确、作者水平较高，教材针对性强，质量较高，本丛书问世7年来，愈来愈得到各校师生的欢迎和好评，至今已发行了240多万册，是国内应用型高校的主流教材之一。2006年被教育部评为普通高等教育"十一五"国家级规划教材，向全国推荐。

由于我国的计算机应用技术教育正在蓬勃发展，许多问题有待深入讨论，新的经验也会层出不穷，我们会根据需要不断丰富本丛书的内容，扩充丛书的选题，以满足各校教学的需要。

本丛书肯定会有不足之处，请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长
《高等院校计算机应用技术规划教材》主编　**谭浩强**

2006年10月1日于北京清华园

前言

本书作为 C++ 的初级入门读物，是作者在总结多年开发、教学经验的基础上编写的，以行之有效的方法让读者快速精通 C++ 。本书以最易于教学的编排和大量附有细致说明的典型程序示例，从基础知识到最新的高级特性，较全面地讲解了 C++ 语言。本书针对学生在学习过程中遇到的困难和提出的问题进行讲解，具有概念清楚、重点突出、难点详尽、便于理解等特点。

通过本书的学习，读者可以了解 C++ 语言及 C++ 程序的一般形式，并逐步掌握 C++ 语言的核心内容，包括引用、重载、类和对象、继承和类的派生、多态性和虚函数、输入/输出流、程序调试等。本书主要针对面向对象程序设计的三大特征——封装性、继承性和多态性做了重点的阐述。第 8 章可以使学生初步了解一些 VC++ 调试工具的使用方法并能进行简单的调试工作，更重要的是可以利用逐步执行程序的调试方法来加深对各知识点的理解。

本书的实用性和可操作性强，提供了大量的编程示例，能够帮助学生透彻理解所学的概念。书中大量的练习题可以使学生进一步巩固所学的内容，确保学生能够真正掌握各章节的内容。

本书既可以作为高等院校计算机及相关专业的专业基础课教材，也可供各类软件开发人员参考。

本书由岳俊梅、李庆义、张峰、薄新全编写。李庆义编写了第 2 章和第 7 章，张峰编写了第 3 章和第 8 章，薄新全编写了第 1 章和第 4 章，岳俊梅编写了第 5 章和第 6 章，最后由岳俊梅统稿。

由于编者水平有限，书中难免有疏漏和不足之处，敬请各位读者批评指正。

编　者

2006 年 12 月

目录

第1章

C++ 语言概述

C 语言和 C++ 语言是当今世界上最流行的编程语言。它们功能强大，应用面广，譬如我们正在使用的 Windows 操作系统的很大一部分，就是使用 C 语言编写成的。

C++ 和 C 的关系如图 1-1 所示。C 是 C++ 的一个子集，不包含 C++ 中的面向对象(OOP)部分。C++ 具有面向对象的优点，成为继 C 语言之后另一个令人不敢忽视的“风云人物”。

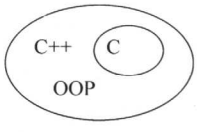


图 1-1　C++ 与 C 的关系图

1.1　认识 C++ 语言

C++ 语言是由 AT&T Bell 实验室的 Bjarne Stroustrup 博士在 20 世纪 80 年代初期提出的。最初的版本叫“C with Classes”(带类的 C)。这表明 C++ 语言在语法上对 C 的主要扩展是对类结构的实现。C++ 的第一个版本于 1979 年 8 月开始在 AT&T 内部使用。同一年晚些时候，开始使用“C++”这个名字。第一个商业 C++ 编译器在 1985 年 10 月发布，同时出版了《C++ Programming Language》的第一版。模板和异常处理是在 20 世纪 80 年代后期被加入的，它们在《The Annotated C++ Reference Manual》和《The C++ Programming Language (2rd Edition)》中被详细描述。

目前的 C++ 是由 ISO C++ 标准定义的，该 C++ 标准(ISO/IEC 14882)在 1998 年以 22 比 0 的投票结果获得批准，并且在《The C++ Programming Language (3rd Edition)》中描述。

但是，标准并不是一个教程，即使是专家级程序员也最好从教科书开始学习 C++ 以及了解C++ 的新特性。

C++ 的新特性如下：

以类定义的形式实现了“对象”，类中不仅包括 C 结构中的数据定义，而且还包括对数据进行操作的函数的声明和定义，这种把数据和函数封装在一个对象中的技术是 C++ 的一个主要革新。

类的实例可以用构造函数和析构函数自动进行初始化和释放，这就消除了程序的初始化错误。

C++ 中类的定义方式增强了数据隐藏性，在默认情况下，类中定义的数据只能被类中的成员函数引用。外部（客户）程序在使用类时不会改变类的内部实现，它们只能通过调用成员函数来访问类。

C++ 允许对操作符和函数进行重载。对一个函数的多个定义可以采用相同的名字，编译器可以在函数调用时识别出合适的定义形式。像“++”和“->”这样的普通操作符也可以被重载为加法的含义。

C++ 允许“类”类型的特征——数据和函数——被子类所继承。子类也称为派生类，子类反过来可以添加更多的数据和函数定义。这就鼓励了用共享类库的方式重用已有代码，从而节省了软件开发过程中的费用。多继承允许派生类从多个基类中继承特征。

C++ 允许类定义虚函数：一个函数有多个定义，在程序进行时决定使用哪个定义。这叫做多态性，它的意思就是在运行时才从函数定义中进行选择，这称为后期绑定（late binding）或动态聚束（dynamic binding）。

可定义模板类，它允许在代码不变的情况下，用不同类型的数据定义同一个类的不同实例。这更进一步提高了代码的重用率。

C++ 中面向对象程序设计的工具是类、继承和虚函数。这些工具使 C++ 语言非常适合编写处理大量相关对象的软件。

和 C 语言相比，C++ 有许多较小的语法改进。它提供了一种新的引用机制，从而对 C 语言中的指针间接引用进行了补充。C++ 简化了动态分配和释放内存的过程，并且实现了一种新 I/O 流库，它用层次分明的类对输入和输出流进行了定义。

C++ 是 C 的扩展，它通过简化设计，使软件能够更好地反映真实世界，降低代码的长度和复杂性，从而增加代码的可靠性。

以上已经对 C++ 的内容做了介绍，让我们开始写第一个程序吧！

1.2 第一个 C++ 程序

C++ 程序的文件扩展名没有统一的标准。在 UNIX 操作系统下，C++ 源代码文件名可以用 .c 或者 .C 结尾。对许多个人计算机系统和工作站来说，其扩展名是 .cpp。一些基于个人计算机的 C++ 编译器要求为 .cxx。本书我们只使用常用的 .cpp。

利用 Visual Studio 可以开发许多各式各样的程序，从数学计算机辅助程序、图像处理程序、特效处理甚至是开发游戏程序等，确实用处很多。以下的例子将通过 Visual Studio 内建的应用程序向导生成一个 C++ 的程序。

步骤 1：打开工程

双击 Microsoft Visual C++ 6.0 软件图标，执行“文件”→“新建”命令，打开 Visual Studio 工程向导。

步骤 2：选择 C++ Source File 选项

在“新建”窗口中打开“文件”选项卡，在清单中选择 C++ Source File 选项，如图 1-2 所示。

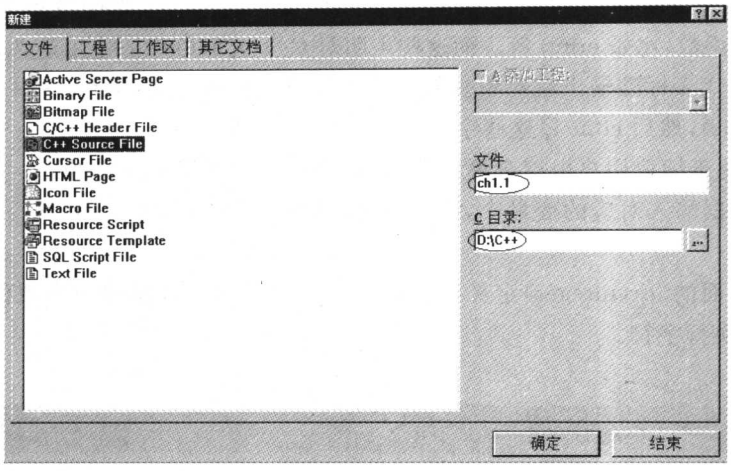


图 1-2　新建 C++ 源文件

步骤 3：生成应用程序

单击“确定”按钮后，生成了应用程序，包含文件、程序代码、注释等。如果不需要加上额外的程序，即可按下键盘上的 F7 键，Visual Studio 将为程序进行编译连接，如图 1-3 所示。

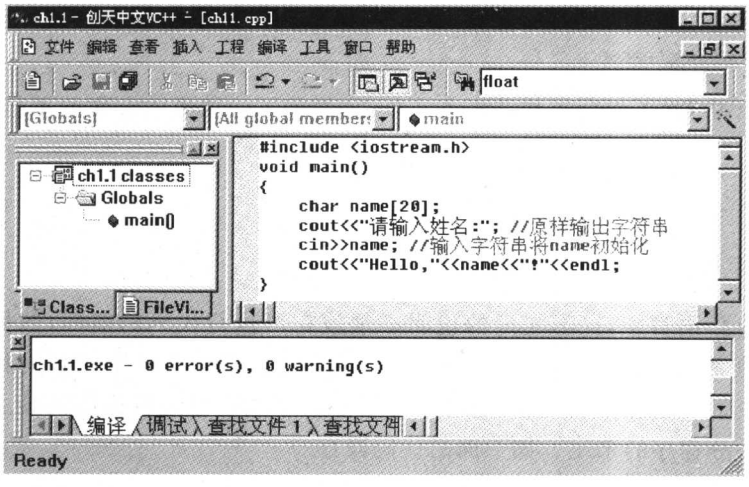


图 1-3　编译和连接结果与 C++ 源文件

现在可以按下 Ctrl+F5 键执行程序了，执行结果如图 1-4 所示。

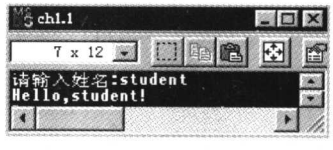


图 1-4　ch1.1 执行结果图

说明：在传统的 C 语言中提供了块注释方式，其形式为：/* … */。而在 C++ 中，除了保留了块注释方式外，还增加了一种更为方便的单行注释方式，其形式为：//…。

iostream.h 是一个标准头文件，它包含了应用程序在进行编译和执行时所需要的声明。iostream.h 和 C 语言中

的 stdio. h 头文件等同，但不可以代替它。iostream. h 声明了 C++ 库函数和工具；stdio. h声明了标准 C 库函数，比如 printf 等。而这些声明对 C++ 流 I/O 库的使用是必要的。

cout 工具代表标准输出流对象。假如终端屏幕是标准输出设备，操作符“<<”右面的字符被送到 cout，然后 cout 将这些字符显示在用户终端上。

cin 工具用于读取用户由键盘输入的数据，直到用户按 Enter 键为止；其中“>>”表示将会依次将数据移入对应的变量中，如果输入的数据包含两个以上的数值，则以空格作为间隔符号。

main()前面的 int(integer)定义当程序执行时向操作系统返回一个数值(一般为 0)。\n 和 endl 为换行字符。

C++ 除了继承了 C 语言中的精华和增加了许多面向对象的特征之外，同时又将 C 语言的不足和问题做了很多改进。下面将总结从 ISO C 到 C++ 中最重要的语法变化，而不讨论 C++ 所增加的面向对象的程序设计工具和 I/O 流、模板以及其他库等。

1. C++ 提供了单行注释方式

在传统的 C 语言中提供了块注释方式，其形式如下：

```
/* explanation sentence */
```

而在 C++ 中，除了保留了块注释方式外，还增加了一种更为方便的单行注释方式，其形式如下：

```
// explanation sentence
```

2. 变量作用域

在传统的 C 语言中，局部变量的说明必须放在可执行代码的前面，数据说明语句和可执行语句的混合将引起编译错误。C++ 提供了(几乎是)在一个函数的任意地方声明并使用一个变量的功能，所说明变量的作用域是从对该变量进行说明的地方开始到该变量所在的最小分程序的末尾。如下面的代码段：

```
#include <iostream.h>
int main()
{
    for(int x = 5; x<10; x++ )
    cout <<"x 的值: "<< x << endl;
    int y = 65;
    cout <<"x 和 y 的值分别为: "<< x << y << endl;
    return 0;
}
```

说明：在C++中，声明语句不一定在函数的开始部分，而是可以跟在其他语句的后面。本例中把变量和初始化当成for循环中控制语句的一部分。

```
#include "iostream.h"
// 首先定义一个全局变量
int temp;
int main(int argc,char * argv[])
{
// 现在再定义一个同名的内部变量
int temp;
// 下面可以使用这个变量
temp = 10;
// 使用全局变量
::temp = 6;
// 在屏幕上显示结果
cout << temp <<"\n"<<::temp <<"\n";
return 0;
}
```

运行后，将在屏幕上输出10，换行，6，换行。

说明：::是变量域运算符，::temp表示引用全局变量中的temp。

3. 关键字

C++引入了许多新关键字，它们包括：class、delete、friend、inline、new、operator、private、protected、public、template、virtual。

这些关键字将在后面进行解释，任何使用这些关键字做变量的程序都是不合法的C++程序。

4. 函数重载

函数重载提供了一个定义和使用变量的方便、强大的功能。在C语言中，如果我们需要一个求两个整数最大值的函数，可以这样写："int Max_Int(int,int);"。如果我们又需要一个求两个浮点数最大值的函数，我们又得定义："float Max_Float(float,float);"。如果又要求其他类型的呢？必须定义不同的其他函数。而C++允许同名函数，只要定义的参数类型或个数不同，编译器会自动进行连接，如int Max(int,int)、float Max(float,float)等。

5. 操作符重载

C++允许对诸如+、=、<<、>>、+=等运算符进行重载，以适应不同的需要。这一特性对于面向对象程序设计来说是非常必要的。在前面的例子中使用了cout << temp << "\n"<<::temp <<"\n"；其中cout是一个在iostream.h中定义的屏幕对象，这个头文件中已经对"<<"进行了重载，使它可以支持诸如整数、浮点数、字符、字符串等常用数据类

型的输出，所以我们可以简单地使用 cout ≪ temp 来输出一个变量而不用指明类型。

6. 默认参数

在 C 语言中，函数调用必须按照参数表全部显式传送。C++ 允许使用默认参数。如已有函数声明 int AFunction(int num＝10)，在调用时可以直接写 AFunction()，编译器会自动认为我们接受了 num＝10 这个参数。这一特性对于参数很多，而大多数一般又无须特殊设置的函数调用显得很方便。

注意：对于多个参数默认的情况，如果某个参数缺省的话，则其后边的参数也只能取默认值。对于设计函数来说，诸如 void function(int a＝0,int b,int c＝2)这样的声明就不那么聪明了。事实上，在这里 a 是不能缺省的。但若改为 void function(int b,int c＝2,int a＝0)就好了。修改后，对于调用函数来说，如果想接受 a 的默认值，那么必须同时接受 c 的默认值。

看以下例子：

```
void function(int a = 0,int b = 1,int c = 2);
{
   ⋮
}
```

以下的函数调用都是合法的。

```
function(); //equal to function(0,1,2)
function(12); //equal to function(12,1,2)
function(12,13); //equal to function(12,13,2)
function(12,13,14);
```

而以下的函数调用则是非法的。

```
function(,13,14);
function(12,,14);
```

7. 按引用传送

C 语言的一些特性有时候不太理想。例如，它在传递参数到函数时，要先把变量复制一份，然后把副本送给函数。一方面，对于较大类型的变量(如结构或对象)，这样传送参数显然效率很低。另一方面，在函数内部只可以修改副本，而无法改变变量本身。在 C 语言中，解决这一问题的方法是使用指针，即按地址传送。C++ 继承了这一特性，而且还引入了“引用”的概念。例如：

```
void Fn(int &nAnotherVar)
{
nAnotherVar = 10;
}
int main()
{
```