普通高等教育“十二五”规划教材

C++程序设计

（第二版）

主　编　杨长兴　刘卫国
副主编　曹岳辉　李利明

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书以程序设计零基础为起点，全面介绍包括面向过程和面向对象的 C++程序设计方法。全书共 10 章，包括 C++基础知识、程序控制结构、函数与编译预处理、数组与指针、自定义数据类型、类与对象、重载与模板、继承与派生、多态性与虚函数、输入输出流。各章节内容由浅入深、相互衔接、前后呼应、循序渐进。

为了提高读者对程序设计思想方法的理解，本书将程序设计语言模型与人类自然语言模型相比较，让读者对程序设计语言模型及其内容的理解有了完整的参照对象。全书各章节选用大量程序设计经典实例来讲解基本概念和程序设计方法，同时配有大量习题供读者练习。本书的配套教材《C++程序设计实践教程》（第二版）提供了本课程的实践内容、上机指导及习题参考答案。

本书语言表达严谨，文字流畅，内容通俗易懂、重点突出、实例丰富。适合作为高等院校各专业程序设计课程的教材，还适合作为广大计算机爱好者的自学参考用书。

本书配有免费电子教案，读者可以从中国水利水电出版社网站以及万水书苑下载，网址为：http://www.waterpub.com.cn/softdown/或 http://www.wsbookshow.com。

图书在版编目（CIP）数据

C++程序设计 / 杨长兴，刘卫国主编. -- 2版. --
北京 : 中国水利水电出版社，2012.1
普通高等教育“十二五”规划教材
ISBN 978-7-5084-9364-0

Ⅰ. ①C… Ⅱ. ①杨… ②刘… Ⅲ. ①
C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2011)第281393号

策划编辑：雷顺加　　责任编辑：李　炎　　封面设计：李　佳

书　名	普通高等教育“十二五”规划教材 C++程序设计（第二版）
作　者	主　编　杨长兴　刘卫国　副主编　曹岳辉　李利明
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（发行部）、82562819（万水）
经　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 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	184mm×260mm　16 开本　19.75 印张　480 千字
版　次	2008 年 1 月第 1 版第 1 次印刷 2012 年 1 月第 2 版　2012 年 1 月第 1 次印刷
印　数	0001—5000 册
定　价	35.00 元

前　言

目前，随着计算机技术的普及与提高，高校计算机基础教学的内容也在不断改革与发展。程序设计课程是大学生必须掌握的计算机基本知识。选用某种程序设计语言作为高校大学生程序设计课程的语言环境，是各校计算机基础教学工作者改革研究的课题之一。过去很长一段时间，许多高校选用 C 语言作为程序设计课程的语言。但随着软件工程技术的不断发展，面向对象的程序设计方法已成为当今软件开发的重要手段之一，尤其是 Visual C++的出现，进一步推动了面向对象与可视化编程技术的发展与应用。因此，掌握面向对象的程序设计方法已经成为大学生计算机应用与软件开发能力的要求之一。由于 C++兼容了 C 语言的功能强、效率高、风格简洁、满足包括系统程序设计和应用程序设计的大多数任务的特点，又扩充了面向对象部分，即支持类、继承、派生、多态性等，解决了其代码的重用问题，C++实际上是既支持面向过程的结构化程序设计又支持面向对象的程序设计的语言。所以，我们根据多年的实际教学经验，在程序设计课程教学改革研究时，选用 C++作为程序设计课程的语言环境。对于本书内容的选择，我们力求面向读者，以程序设计零基础为起点，全面介绍了包括面向过程和面向对象的 C++程序设计方法。让读者首先接受面向对象的程序设计的思想方法，并理解面向对象的程序设计是需要以面向过程的程序设计方法作为基础的。

全书共分为 10 章，第 1 章介绍 C++的基础知识；第 2 章介绍程序控制结构；第 3 章介绍函数；第 4 章介绍数组与指针；第 5 章介绍自定义数据类型；第 6 章介绍类与对象；第 7 章介绍重载与模板；第 8 章介绍继承与派生；第 9 章介绍多态性与虚函数；第 10 章介绍输入输出流。从全书组织结构来看，首先定位 C++是兼顾面向过程和面向对象的程序设计语言，面向对象的程序设计是以面向过程的程序设计为基础的。因此，在第 1～5 章以介绍面向过程的程序设计为主；在第 6～10 章以介绍面向对象的基本思想与方法为主。

本书编者长期从事程序设计课程的教学工作，并利用 C/C++/Visual C++开发了许多软件项目，具有丰富的教学经验和较强的科学研究能力。编者本着加强基础、注重实践、突出应用、勇于创新的原则，力求使本书达到有较强的可读性、适用性和先进性。我们的教学理念是：教学是教思想、教方法，真正做到“授人以鱼，不如授人以渔”。为了加强读者对程序设计思想方法的理解，本书将程序设计语言模型与人类自然语言模型相比较，让读者对程序设计语言模型及其内容的理解有了完整的参照对象。为了提高读者的编程技巧，选用了大量的经典例题，这些例题与相应章节的基本内容是完全吻合的，而且读者对这些例题的自然解法是相当熟悉的。例题还备有多种可能的解答，以期拓展读者的解题思路。为了便于读者自学，在全书的内容组织、编排上注重由浅入深、深入浅出、循序渐进。因此，本书适合作为高等院校各专业程序设计课程的教材，也适合作广大计算机爱好者的自学参考用书。如教师选用本书作为大学生程序设计课程的教材，可根据实际教学课时数调整或取舍内容。

本书所给出的程序示例均在 Visual C++ 6.0 环境下进行了调试和运行。为了帮助读者更好地学习 C++，编者还编写了配套教材《C++程序设计实践教程》（第二版）一书，该配套教材提供了本课程的实践内容、上机指导及习题参考答案。

本书由杨长兴、刘卫国任主编，负责全书的总体策划、统稿和定稿工作。曹岳辉、李利明任副主编，协助主编完成统稿、定稿工作。各章编写工作分工如下：第 1 章由杨长兴编写，第 2 章由周春艳编写，第 3 章由朱从旭编写，第 4 章由周欣然编写，第 5 章由李小兰编写，第 6 章由刘卫国编写，第 7 章由曹岳辉编写，第 8 章由吕格莉编写，第 9 章由李利明编写，第 10 章由蔡旭晖编写。

本书的编写得到了中南大学信息科学与工程学院施荣华等教授的大力支持与帮助，在此表示衷心的感谢。在本书的编写过程中，王小玲、严晖、周肆清、舒卫真、田琪、邵自然、罗芳、杨莉军等老师参与了大纲的讨论和文稿校对工作，本书吸收了他（她）们许多宝贵的意见，在此一并表示感谢。编者在编写本书的过程中参考了大量的文献资料，在此也向这些文献资料的作者表示衷心感谢。

由于本书编者水平所限，成稿时间仓促，书中如有疏漏及不妥之处，敬请读者不吝赐教。

编 者

2011 年 12 月

目　　录

第 1 章　C++基础知识

通过本章的学习，掌握计算机程序设计语言模型；掌握 C++程序设计语言集成开发环境；掌握 C++应用程序开发过程；掌握 C++数据类型、常量、变量、运算符与表达式的运用；掌握运算符的优先级和结合性、表达式的类型转换；掌握变量的存储类型、作用域与生存期的概念。

- 计算机程序设计语言模型；
- 各种表达式的运用；
- 变量的存储类型、作用域与生存期。

计算机程序是解决任何实际问题的基础。学习 C++语言程序设计的目的，就是要学会使用 C++语言编写出适合自己实际需要的程序。程序包括数据和施加于数据的操作两个方面的内容。数据是程序处理的对象，操作步骤反映了程序的功能。不同类型的数据有不同的操作方式和取值范围，程序设计需要考虑数据的表示以及操作步骤（即算法）。C++语言具有丰富的数据类型和相关运算。本章首先介绍程序设计的基本概念、C++程序的基本结构以及 C++程序的执行步骤，然后介绍 C++的数据类型和数据的运算，这些内容是以后学习的基础。

1.1　程序设计语言的基本概念

计算机程序设计语言是人类在计算机上解决实际问题的一种编码规则工具。当一个求解问题能够用数学模型表达时，人们会考虑用某种程序设计语言将该问题的数学模型表示成计算机可以接受的程序形式，再由计算机自动处理这个程序，生成人们所需要的结果。

程序设计语言随着计算机科学的发展而发展，它由最早的机器语言形式逐步发展成为现在的接近人类自然语言的形式。

20 世纪 50 年代的程序设计是使用机器语言或汇编语言编写的，用这样的程序设计语言设计的程序相当烦琐、复杂，不同机器使用的机器语言或汇编语言几乎完全不同。能够使用这类语言编写程序的人群极其有限，也就限制了这类计算机程序设计语言的普及和推广，必然影响了计算机的普及和应用。

20 世纪 50 年代中期研制出来的 FORTRAN 语言是计算机程序设计语言历史上的第一个高级程序设计语言。它在数值计算领域首次将程序设计语言以接近人类自然语言的形式呈现在人

们的面前，它引入了许多目前仍在使用的程序设计概念，如变量、数组、分支、循环等。20世纪50年代后期研制的ALGOL语言进一步发展了高级程序设计语言，提出了块结构的程序设计概念。即一个问题的求解程序可以由多个程序块组成，块与块之间相对独立，不同块内的变量可以同名，但互不影响。

到了20世纪60年代后期，人们设计出来的程序越来越庞大，随之而来的问题是程序越庞大，程序的可靠性越差，错误越多，并且难以维护。程序设计人员难以控制程序的运行，这就是当时的“软件危机”问题。为了解决“软件危机”问题，荷兰科学家E.W.Dijkstra在1969年首次提出了结构化程序设计的概念，这种思想强调从程序结构和风格上研究程序设计方法。后来，瑞士科学家Niklans Wirth的“算法+数据结构=程序”思想进一步发展了结构化程序设计方法，将一个大型的程序分解成多个相互独立的部分（称为模块）。模块化能够有效分解大型、复杂的问题，同时每个模块相互独立，提高了程序的维护效率。这就是面向过程的结构化程序设计思想。所谓面向过程的结构化程序设计思想是人们在求解问题时，不仅要提出求解的问题，还要精确地给出求解问题的过程（将问题的求解过程分解成多个、多级相互独立的小模块）。20世纪70年代初面世的C语言就是典型的、面向过程的结构化程序设计语言。

面向过程的结构化程序设计是从求解问题的功能入手，按照工程的标准和严格的规范将求解问题分解为若干功能模块，求解问题是实现模块功能的函数和过程的集合。由于用户的需求和硬件、软件技术的不断发展变化，按照功能划分将求解问题分解出来的模块必然是易变和不稳定的。这样开发出来的模块可重用性不高。20世纪80年代提出的面向对象的程序设计方法即是为了解决面向过程的结构化程序设计所不能解决的代码重用问题。面向对象的程序设计方法是从所处理的数据入手，以数据为中心而不是以求解问题的功能为中心来描述求解问题。它把编程问题视为一个数据集合，数据相对于功能而言，具有更好的稳定性。这就是“对象+对象+……=程序”的理论。面向对象的程序设计与面向过程的结构化程序设计相比，最大的区别就在于：前者关心的是所要处理的数据，而后者关心的是求解问题的功能。面向对象的程序设计方法很好地解决了“软件危机”问题。

面向对象的程序设计语言有两类：一类是完全面向对象的语言，另一类是兼顾面向过程和面向对象的混合式语言。C++语言就是后一类的典型代表。

C++由AT&T公司贝尔实验室的Bjarne Stroustrup博士开发。它兼容了C语言的一切特征，以C语言为核心，扩充了面向对象部分。它保留了C语言功能强、效率高、风格简洁、满足包括系统程序设计和应用程序设计的大多数任务的特点。过去在C语言环境下编写的程序几乎无需改动就可以在C++环境下编译运行。另外，C++支持面向对象的程序设计，它支持类、继承、派生、多态性等，因此解决了代码的重用问题。所以C++是既支持面向过程的结构化程序设计又支持面向对象的程序设计的语言。

从20世纪80年代开始，C++逐步成为人们所喜爱的面向对象的程序设计语言开发工具。众多公司纷纷开发C++编译程序或开发环境。典型的开发环境有Borland公司开发的Turbo C++和Borland C++、Microsoft 公司的Microsoft C/C++等。上面提到的C++语言版本都需要DOS环境的支持，或需要在Windows的DOS模式下运行。而Microsoft 公司开发的Visual C++则是引进了Windows MFC（Microsoft Foundation Class）类库的基于Windows环境的可视化集成开发环境，它是将编辑、编译、链接、调试运行集成于一体的开发环境。软件版本有：1.X、2.0、4.0、5.0、6.0等。本教程使用的软件环境是Visual C++ 6.0。

1.2　C++程序的基本结构

学习一种语言之初了解其程序结构是必要的，本节的目的是让读者对 C++语言有一个整体的、框架性的认识。

读者可以了解一下一般程序设计语言的模型。通过图 1-1 来进行理解：学习 C++程序设计，主要是学习（见图 1-1）根据词法规则用 C++字符集中的字符构造单词；根据语法规则用单词构造语句；根据逻辑规则（任务内在的联系）用语句构成函数（程序）。读者可以根据图 1-1 自学其他程序设计语言。实际上，任何语言都遵从这种模式，包括自然语言（英语、汉语等）。对于自然语言，只是字符集中的字符多，构词规则复杂，语法规则更复杂，由若干语句组成的集合叫文章或文章段落而已。其实计算机程序设计语言就是从自然语言模型中简化出来的。理解了这个道理，对于学习程序设计语言是很有帮助的。

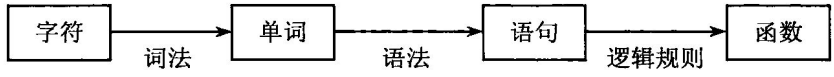

图 1-1　程序设计语言结构图

1.2.1　C++程序结构

一般来说，C++程序的结构包含声明区、函数区两个部分，在任何一个区内都可以随时插入程序的注释。下面通过一个简单的例子来认识 C++程序的基本结构。

【例 1.1】从键盘输入圆的半径，求圆的面积。

程序代码如下：

```
//*****ex1_1.cpp*****
#define PI 3.14159
#include <iostream>
using namespace std;
float sum(float x);
void main()
{
    float r,s;
    cout<<"Input r:";
    cin>>r;
    s=sum(r);
    cout<<"r="<<r<<" "<<"s="<<" "<<s<<endl;
}
float sum(float x)
{
    return PI*x*x;
}
```

要说明的是，本章的例子中使用了标准输入输出流语句和其他语句。关于这些语句的详细使用方法，请参阅第 2 章的有关内容。

1. 声明区

声明区在函数之外。程序的声明区可能需要编写：

（1）宏定义，如例 1.1 中的#define PI 3.14159；

（2）包含文件，如例 1.1 中的#include <iostream>；

（3）函数声明，如例 1.1 中的 float sum(float x)；

（4）条件编译；

（5）全局变量声明；

（6）结构体等的定义；

（7）类的定义。

2. 函数区

一个程序由一个主函数 main()和多个（可以是 0 个）其他函数组成。每个函数都是由函数声明部分与函数体部分组成。程序的执行从 main()函数开始。

函数声明部分包括函数返回值类型、函数名、函数的形式参数。如在例 1.1 中，有两个函数：float sum()和 void main()，其中 main()的函数返回值类型是 void（没有返回值类型），函数名就是 main，函数没有形式参数；sum() 的函数返回值类型是 float，函数名为 sum，函数的形式参数是 float x。

函数体部分是用一对花括号{}括起来的完成该函数所表达的功能的语句的集合。语句可以是数据描述语句或数据操作语句。数据是操作的对象，操作的结果将会改变数据的状态。对于一个实际应用问题，首先要考虑用什么形式表达问题，也就是用什么数据结构表达问题。在具体程序中用数据描述语句表达数据结构，也就是用数据类型表达数据。有了数据结构，再考虑如何操作数据，也就是用数据操作语句实施为解决问题所提出的算法，产生所需要的结果。所以著名计算机科学家 Niklans Wirth 提出了如下公式：

程序=数据结构+算法

一个程序用来实现某项任务，一个任务可以分解为多个子任务，每个子任务还可以进一步分解成更小的子任务，每个子任务可以用一个函数或多个函数来表达。在 C++中，一个程序的某些函数可以存放在一个文件中，而另外一些函数可以存放在另一个文件中。所以一个 C++程序的组装形式是：

函数→文件→程序

在 C++程序中，各函数之间只有调用关系，即组成 C++程序的若干函数之间是“平等”关系。在一个函数中不能定义另一个函数，即函数之间不存在嵌套关系。在一个 C++程序中，不被任何函数调用的函数是无用的。

函数体内有若干条语句用来实现函数的功能。每条语句由单词（常量、变量、关键字等）构成，而单词由字符构成。

1.2.2 C++程序的书写格式

C++程序的书写格式比较灵活，书写程序时可以任意换行，一行内可以书写多条语句，一条语句可以书写在多行上，只要每条语句以分号（;）结束即可。也因为如此，所以 C++程序可读性差，为了提高程序的可读性，C++程序的书写格式有如下约定：

（1）C++程序中，每行一般书写一条语句；语句较短时，多条语句可书写在一行内。语句较长时，一条语句可写在多行上。

（2）C++程序中，每条语句以分号结束，表示一条语句的结束，但函数说明行和声明区的多数语句后不用分号。语句前面没有标号，只有 goto 语句的转向目标语句前加标号。

（3）C++程序中，使用向右缩进方法表达程序中的层次结构，如花括号{}内的函数体、循环语句的循环体、if 语句的 if 体和 else 体一般都向右缩进几个字符。花括号是函数体或复合语句的定界符。

（4）C++程序中，可使用多行注释或单行注释以增强程序的可读性。多行注释以“/*”开始，以“*/”结束，占据多行。单行注释以“//”开始，占据一行。

1.3　一个应用程序的开发过程

对于初次学习程序语言的读者来说，了解和掌握某种程序设计语言的开发环境和应用程序开发过程是十分重要的。本节介绍 Visual C++ 6.0 开发环境及开发一个应用程序的过程。

1.3.1　Visual C++ 6.0 开发环境

Visual C++ 6.0 是 Microsoft 公司于 1998 年推出的基于 Windows 9.X 和 Windows NT 的可视化开发环境。它为用户提供一个集编辑、编译、链接、调试运行于一体的集成环境，即在一个程序的控制下可以分步或一次性地完成编辑、编译、链接、调试运行工作。

C++程序的实现与其他高级语言程序一样，也需要经历编辑、编译、链接、调试运行 4 个步骤。

编辑程序完成对 C++程序的输入、修改。以文件形式存储的 C++程序称为 C++源程序，文件的扩展名为.cpp。除了使用编译系统提供的编辑程序外，还可以使用如 Word、记事本等常规编辑程序编辑 C++源程序，但必须以.cpp 为扩展名存储。

编译程序把包括 C++源程序在内的项目翻译成目标程序，即生成与 C++源程序相对应的目标程序。目标程序的文件扩展名是.obj。编译程序一般会生成与源程序同名的目标程序。

链接程序把目标程序链接成可执行程序，可执行程序的文件扩展名是.exe，生成可执行程序时需要编译系统提供的库文件支持。链接程序一般会生成与源程序同名的可执行程序文件（exe 文件）。

上述三步无误时，即可直接调试运行可执行程序。

1.3.2　一个应用程序的开发过程

下面通过以 AppWizard（应用程序向导）创建一个控制台应用程序（基于 DOS 平台的应用程序）为例向读者介绍应用程序的开发过程。有关 Visual C++ 6.0 开发环境和 Windows 应用程序的开发过程请参考配套的实践教程的相关章节。

1. *启动 Visual C++ 6.0 开发环境*

在安装 Visual C++ 6.0 后，启动 Visual C++ 6.0，用户可以看到如图 1-2 所示的 Visual C++ 6.0 开发环境界面。

2. *创建工程项目*

开发一个新的程序就是建立一个新的工程项目。所谓工程项目就是所开发的应用程序的一切资源的集合。它包含了应用程序的源程序代码、类、资源等。过程如下。

在图 1-2 的“文件”菜单下选择“新建”命令，弹出“新建”对话框，在“新建”对话框

中选择“工程”标签，如图 1-3 所示。从图 1-3 可以看出，在“工程”选项卡中可以建立多种用途的工程项目，如选择“Win32 Console Application”选项可以建立一个基于控制台应用程序的项目文件，而选择“MFC AppWizard(exe)”选项可以建立一个基于 Windows 平台应用程序的项目文件，等等。为了建立一个名为“ex1_1.dsw”的项目文件，需要在图 1-3 中的“位置”文本框中设置项目文件存放的位置，在“工程名称”文本框中输入项目文件名。同时选择“平台”文本框中的“Win32”选项（指明开发的项目是 Windows 32 位环境）。图 1-3 是补充了上述信息的“工程”选项卡。

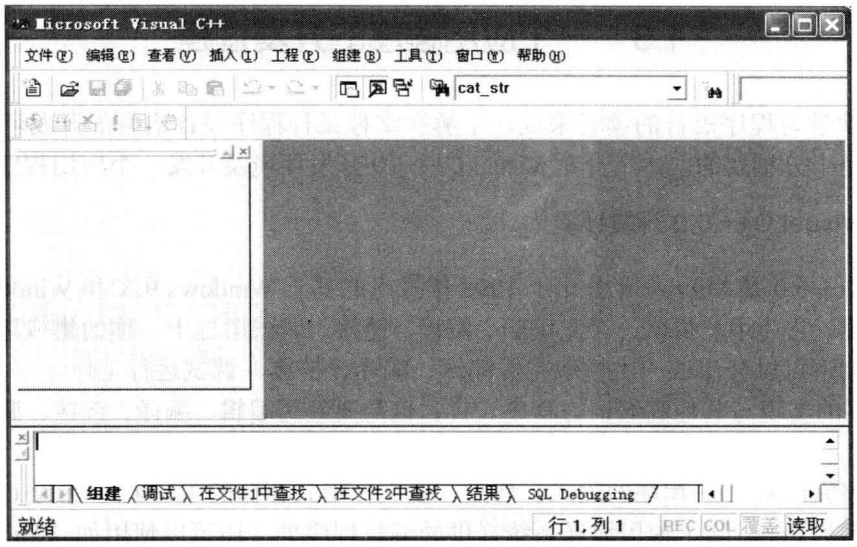

图 1-2 Visual C++ 6.0 开发环境界面

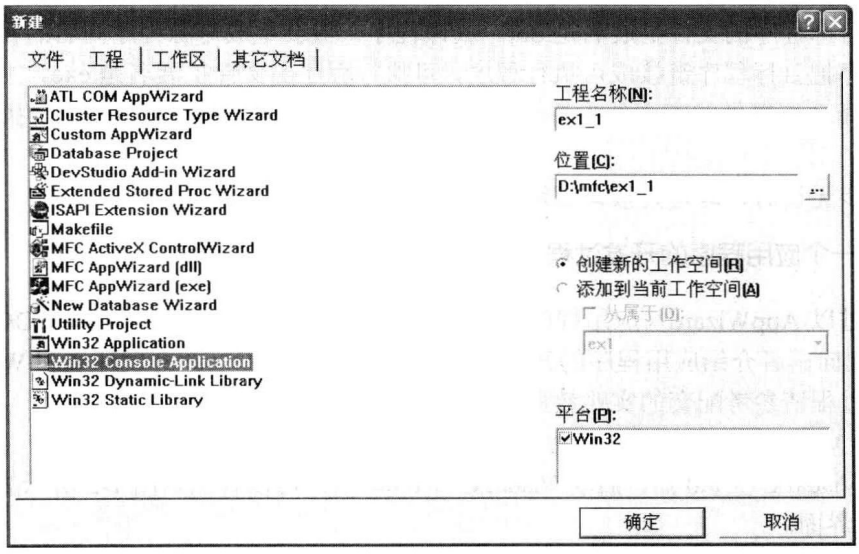

图 1-3 “新建”对话框中“工程”选项卡

单击图 1-3 中的“确定”按钮，弹出如图 1-4 所示的创建“Win32 Console Application－步骤 1 共 1 步”对话框。

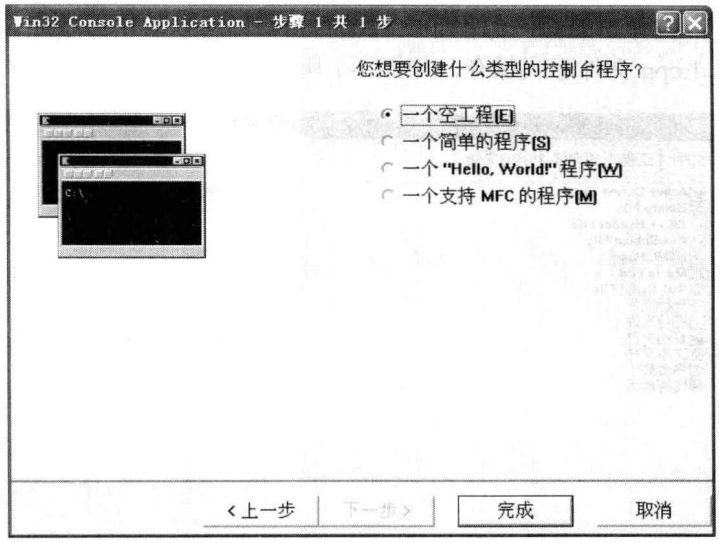

图 1-4　“Win32 Console Application”对话框

在图 1-4 中，给出了针对 Win32 Console Application 的 4 种类型，Visual C++ 6.0 开发环境会根据不同的类型为所创建的项目文件自动增加相应的功能。如果选择“一个空工程”单选按钮，则生成一个空白的项目。单击“完成”按钮，即可完成新项目的创建，生成一个扩展名为.dsw 的项目文件。

3. 打开项目文件

打开一个已存在的项目文件，可以通过图 1-2 的“文件”菜单下的“打开工作区”命令实现。选择“打开工作区”命令，弹出如图 1-5 所示的“打开工作区”对话框，选择所需的文件即可打开项目文件。

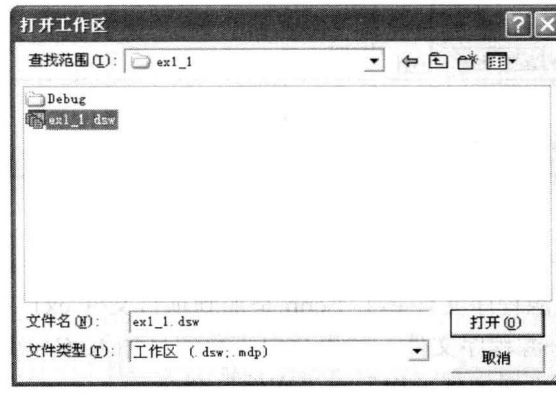

图 1-5　“打开工作区”对话框

当然，还可以使用“文件”菜单下的“打开”命令或工具栏上的“打开”按钮打开项目文件。这两种打开方式可以打开如 C++源程序文件、C++头文件、资源文件等任何类型的文件。

4. 创建 C++源程序文件

在图 1-2 的“文件”菜单下选择“新建”命令，弹出“新建”对话框，在“新建”对话框中选择“文件”标签，选择“C++ Source File”选项，并指出要创建的 C++源程序文件名及其所在的位置，选择“添加到工程”复选框。“新建”对话框中“文件”选项卡如图 1-6 所示。

单击“确定”按钮，在弹出的C++源程序文件编辑窗口即可以编辑源程序。我们将本章例1.1源程序代码以ex1_1.cpp文件形式保存，如图1-7所示。

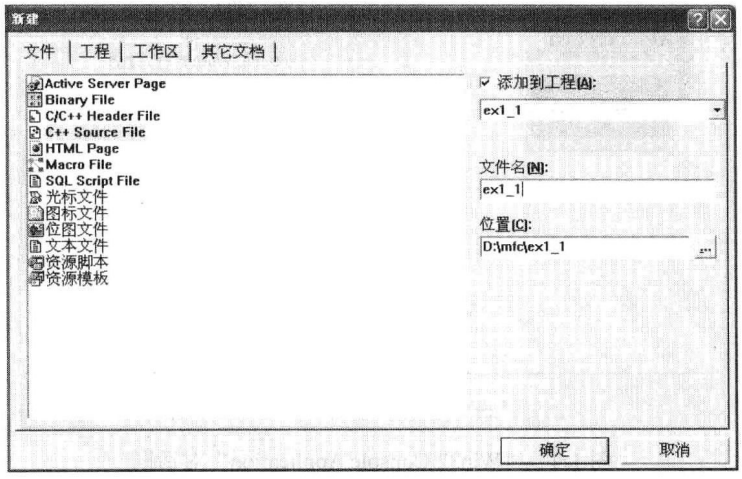

图1-6 “新建”对话框中“文件”选项卡

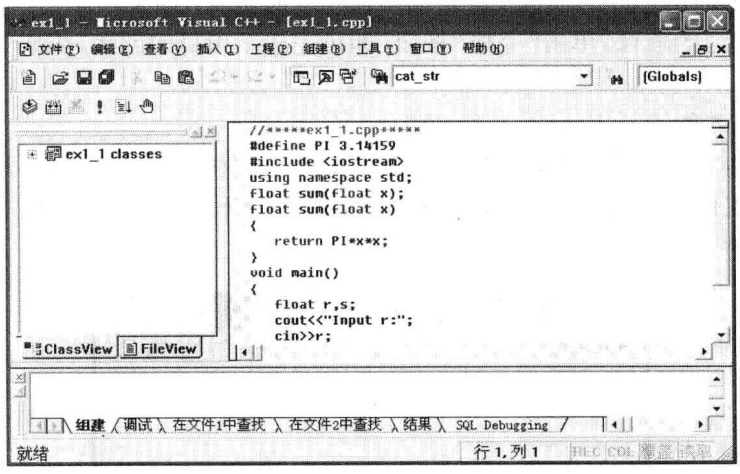

图1-7 添加了源程序代码的项目窗口

因为在此之前已打开前面创建的项目文件ex1_1.dsw，所以在图1-6中，只要选择“添加到工程”复选框，就表明源程序文件ex1_1.cpp是加到项目文件ex1_1.dsw中的。当然可以向项目文件再添加多个C++源程序文件。如果是向当前项目加入C++源程序文件，则在图1-2中的“工程”菜单下选择“增加到工程”→“文件”命令。

至此，项目文件ex1_1.dsw的创建完成。

5. 编译、链接与运行程序

在Visual C++ 6.0的开发环境中，编译、链接与运行3个步骤可以各自单独进行（按照编译、链接再运行的顺序），也可以将编译与链接合二为一进行，当然也可以将编译、链接运行一步完成。

在图1-7中，在“组建”菜单下选择“编译”进行编译，当然可以在工具栏单击“编译”按钮，或按下组合键Ctrl+F7。在“组建”菜单下选择“组建”进行编译与链接（在没有编译

的情况下），或单击工具栏上的按钮或按下 F7 键。在“组建”菜单下选择“执行”或单击工具栏上的按钮或按下组合键 Ctrl+F5 进行编译、链接和运行（在没有编译、链接的情况下）。

图 1-8 即为编译、链接完成后的输出窗口。当编译或链接过程中出现错误时，会在输出窗口提示错误信息，用户可根据错误提示信息对源程序进行修改，直至没有错误信息提示。

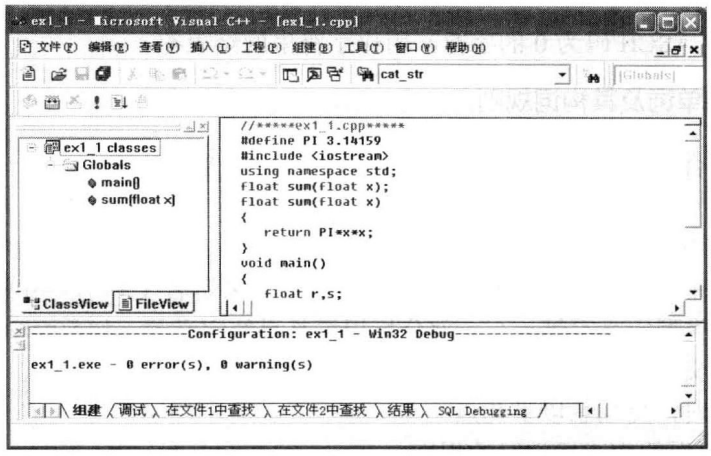

图 1-8　编译链接完成后的输出窗口

例 1.1 程序所建立的项目的运行结果如图 1-9 所示。

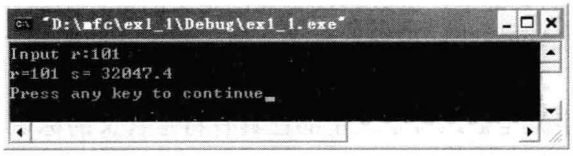

图 1-9　例 1.1 程序运行结果

1.4　C++的词法规则

某种程序设计语言在定义常量、变量类型时，必须对常量、变量命名。常量、变量的名字就像自然语言英语的单词一样，这样的常量名、变量名实际上是这种程序设计语言中的一个个单词。常量名、变量名可以定义、理解成表示名字的标识符。除了常量、变量之外，某种程序设计语言还有关键字、运算符、分隔符等基本单词，这些单词的构成（书写规则或命名）都必须遵守该语言的词法规则。

本节介绍的就是 C++中各种单词的构词规则。请读者结合图 1-1 学习本节内容并思考本节内容属于图 1-1 的哪个环节。

由一种语言提供的字符集中的字符构造的单词必须遵守该语言的词法规则。在一种语言中，单词涉及标识符、关键字、运算符、常量、变量、注释符、分隔符等。C++也不例外。本节只讨论 C++中的基本词法规则。

1.4.1　C++的字符集组成

C++的字符集包括如下字符：

（1）26 个小写字母：a～z。

（2）26 个大写字母：A～Z。

（3）10 个数字：0～9。

（4）标点和特殊字符：+、_、*、/、,、:、;、?、\、”、’、～、|、!、#、%、&、(、)、[、]、{、}、^、<、>空格。

（5）空字符：ASCII 码为 0 的字符，用作字符串的结束符。

1.4.2 C++的单词及其构词规则

单词由若干个合法字符组成，下面介绍一些 C++中的常用单词。

1. 标识符

标识符是由字母、下划线和数字组成的字符序列，第一个字母必须是字母或下划线，不能是数字。标识符中的字母大小写是不同的。标识符用来命名 C++程序中的常量、变量、函数名、语句标号及类型定义符等。有一部分标识符是系统定义的，如前面学习过的 cin、cout，有一部分是用户定义的。本小节关心的是用户定义的标识符。

在定义标识符时，要注意以下几点：

（1）要遵守上面的构成标识符的规则。

ABC、XYZ123、A_Y、ycx11、_name 是合法标识符。而 5xyz、m.x、!abc、x-y 是非法标识符（数字“5”不能出现在标识符开头，“.”、“!”、“-”不能出现在标识符中）。

（2）系统已经使用的关键字、函数名或其他已定义的单词不能再定义成标识符。

（3）定义标识符时尽可能让标识符有意义，便于阅读，即做到“见名知意”。

2. 关键字

关键字（又称保留字）是被系统定义了的已具有特定含义的标识符。下面给出了 C++中的关键字。要说明的是：ANSI C 规定了 32 个关键字，ANSI C++在此基础上增加了一些（黑体字部分），某个实现的版本可能还会增加一些关键字。

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while	**bool**	**catch**	**class**	**const_cast**
delete	**dynamic_cast**	**explicit**	**false**	**friend**	**inline**
mutable	**namespace**	**new**	**operator**	**private**	**protected**
public	**reinterpret_cast**	**static_cast**	**template**	**this**	**throw**
true	**try**	**typeid**	**typename**	**using**	**virtual**
_asm	_far	_fotran	_huge	_near	_pascal

3. 运算符

运算符与操作数连接组成表达式。连接一个操作数的运算符称为单目运算符，连接两个操作数的运算符称为双目运算符，连接三个操作数的运算符称为三目运算符。有关运算符的种类、功能、优先级和结合性等问题将在 1.7 节中详细介绍。