普通高等教育“十二五”规划教材

C++程序设计教程

主　编　张晓如　华　伟
副主编　祁云嵩　王　芳　王　勇

科学出版社

普通高等教育“十二五”规划教材

C++程序设计教程

主　编　张晓如　华　伟
副主编　祁云嵩　王　芳　王　勇

科学出版社
北京

内 容 简 介

本书系统地讲述了 C++语言的基本知识，并详细介绍了面向过程程序设计和面向对象程序设计的基本方法。全书共 11 章，内容包括概述、数据类型与表达式、流程控制语句、数组、函数、结构体与简单链表、类和对象、继承与多态性、友元函数与运算符重载、模板和异常处理及输入/输出流。

本书可作为高等院校 C++语言程序设计的教材，也可作为程序设计爱好者的参考用书。

图书在版编目(CIP)数据

C++程序设计教程 / 张晓如，华伟主编. —北京：科学出版社，2013
普通高等教育“十二五”规划教材

ISBN 978-7-03-038385-3

I. ①C… II. ①张… ②华… III. ①C 语言–程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 189876 号

责任编辑：相 凌 王迎春 / 责任校对：鲁 素
责任印制：肖 兴 / 封面设计：华路天然工作室

科 学 出 版 社出版
北京东黄城根北街 16 号
邮政编码: 100717
http://www.sciencep.com

北京通州皇家印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2013 年 8 月第 一 版 开本：787×1092 1/16
2013 年 8 月第一次印刷 印张：15 1/4
字数：400 000

定价：34.00 元

(如有印装质量问题，我社负责调换)

前　言

C++语言作为程序设计的经典语言，功能强大，广泛应用于程序开发。为适应应用型人才培养的需求，各高校普遍开设C++程序设计课程。各种类型的教材应运而生，虽然各有特点，但要找到一本既适合教师教学需要，又能满足学生学习需求的教材并不容易。根据近年来的教学研究成果，结合教学改革实践和计算机程序语言精品课程建设，作者编写了本书。本书叙述简明易懂，借助典型例题深入浅出地分析了C++的主要概念、语法以及编程方法，力图帮助教师在有限的教学时间内，向学生阐述C++语言程序设计的核心内容，使学生更好地掌握C++语言程序设计的精髓。

本书共11章，第1章为程序设计的概述，第2~6章主要阐述面向过程程序设计方法，第7~11章阐述面向对象程序设计方法。本书内容重点突出，语言表达精练，每章都配有复习巩固用习题。全书以最简洁的语言表达C++程序设计的主要特征，所有程序均在VC++6.0中通过编译，运行结果正确。

为配合课程内容的学习，本书还同时配有《C++程序设计习题与实验教程》、《C++程序设计实践教程》，它们既相互关联，又能独立使用，起到相辅相成的作用。

本书在编写过程中，参阅了部分教材及资料，在此向各位作者表示衷心的感谢。本书出版过程中，学校各级领导、教材科老师、出版社编辑都做了大量工作，付出了辛勤的劳动，在此一并表示感谢。

由于编者水平有限，书中尚存不足和疏漏之处，恳请广大读者批评指正，以便能进一步做好改进完善工作。

编　者

2013年6月

目　录

第1章 概　述

1.1 程序设计语言

1.1.1 程序设计语言概述

计算机程序是人们为解决实际问题而需要计算机完成的一系列操作指令的有序集合。程序设计语言是人与计算机交流的工具，是计算机可以识别的语言，具有特定的词法与语法规则。一种计算机程序设计语言能够使程序员准确地定义计算机所需要使用的数据，在不同情况下应当采取的操作。计算机语言的种类非常多，总的来说可以分成机器语言、汇编语言、高级语言3大类。目前常用的编程语言有两种形式：汇编语言和高级语言。

机器语言是直接用二进制代码指令表达的计算机语言，指令是用0和1组成的一串代码。例如，将100与200相加的机器语言程序由以下两条指令实现：

```
1101 1000 0110 0100 0000 0000
0000 0101 1100 1000 0000 0000
```

虽然机器语言能被计算机直接识别和执行，但对于人类来说十分晦涩难懂，更难记忆与编写。在计算机发展初期，只能用机器语言编写程序，在这一阶段计算机编程语言与人类的自然语言之间存在巨大的鸿沟，软件开发难度大，周期长，修改维护困难。

为了解决机器语言编程的困难，程序员使用类似英文缩写的助记符来表示指令，从而产生了程序设计的汇编语言(assembly language)，如使用ADD、SUB助记符分别表示加、减运算指令。将100与200相加的汇编语言如下：

```
MOV AX, 100
ADD AX, 200
```

机器不能直接识别使用汇编语言编写的程序，而需要由汇编程序将汇编语言翻译成机器语言。虽然汇编语言比机器语言便于理解和编程，但仍然与人类的自然表达方式相差甚远。汇编语言实质上仍是机器语言，同样属于低级语言。

为了进一步方便编程，人们开发了更加接近人类自然语言习惯的高级语言，程序使用了更有意义和容易理解的语句，更容易描述具体的事物与过程，编程效率大大提高。例如，仍然是将100与200相加，用高级语言C++可描述如下：

```
100+200
```

高级语言与计算机的硬件结构及指令系统无关，它有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且容易学习。但高级语言编译生成的程序代码一般比用汇编语言设计的程序代码要长，执行的速度也更慢。

高级语言是目前绝大多数编程者的选择。与汇编语言相比，它不但将许多相关的机器指令合成单条指令，并且去掉了与具体操作有关而与完成任务无关的细节。例如，使用堆栈、寄存器等，可以大大简化程序中的指令。同时，由于省略了很多细节，编程者不需要有太深

奥的专业知识。

高级语言主要是相对于汇编语言而言的，它并不是特指某一种具体的语言，而是包括了很多编程语言，如 C++、Visual Basic(VB)、Java 等，这些语言的语法、命令格式各不相同。

学习一种计算机程序设计语言，不仅是为了掌握一种实用的计算机软件设计工具，更重要的是掌握程序设计的基本方法，培养良好的思维习惯，提高分析问题、解决问题的能力，为今后的学习和工作打下良好的基础。

1.1.2 C++程序设计语言

C++语言是目前应用最为广泛的计算机程序设计语言之一。C++由 C 语言扩充、改进而来。当 C 语言发展到顶峰的时刻，出现了一个版本叫 C with class 的版本，这就是 C++的雏形，它在 C 语言中增加了 class 关键字和类。程序设计者都希望在 C 语言中增加类的概念，后来 C 标准委员会决定以 C 语言中的++运算符来体现 C 语言的进步，因此叫 C++，并成立了 C++标准委员会。虽然 C++是作为 C 语言的增强版出现的，但可以认为它是一门独立的语言；C++并不依赖 C 语言，初学者可以完全不学习 C 语言，而直接学习 C++语言。

C++程序设计语言具有下列特点。

(1)C++完全兼容 C，具有 C 语言的简洁、紧凑、运算符丰富，可直接访问机器的物理地址，使用灵活方便，程序书写形式自由等特点。大多数 C 语言程序代码略作修改或不作修改即可在 C++集成环境下运行。

(2)C++作为一种面向对象的程序设计语言，它使程序的各个模块间更具独立性，程序的可读性更好，代码结构更加合理，设计和编制大型软件时更为方便。

(3)用 C++语言设计的程序可扩充性更强。

目前很多软件支持基于 C++语言的程序开发，例如，Microsoft Visual C++、Borland C++、Watcom C++、Tubor C++等。本书中的程序都是采用 Microsoft Visual C++ 6.0 编写的，本书将在 1.3 节详细介绍如何在 VC6.0 环境下创建并执行 C++应用程序。

1.2 程序设计思想

计算机对问题的求解方式通常可以用数学模型抽象。随着社会与科学技术的发展，人们要求计算机处理的问题越来越复杂，计算机研究人员不断寻求简捷可靠的软件开发方法。程序设计的方法主要有两类：面向过程的程序设计和面向对象的程序设计。

1.2.1 面向过程的程序设计

面向过程的程序设计是一种自上而下的设计方法。例如，在 C 语言中，程序员用一个 main 函数概括整个应用程序需要完成的工作，而 main 函数由一系列子函数的调用组成。main 函数中调用的每个函数又都可以再被分解成更小的子函数，重复这个过程，就可以完成一个过程式的设计。其特征是以函数为中心，函数作为划分程序的基本单位，数据在过程式的程序设计中往往处于从属地位。

一个典型的过程式程序结构图如图 1-1 所示。图中的箭头代表函数间的调用关系，也是函数间的依赖关系。main 函数依赖于其子函数，这些子函数又依赖于各自的子函数。由于面向过程的程序设计采用的是一种自顶向下、逐步求精的方法，据此可以设计出结构良好、易

于调试的程序。

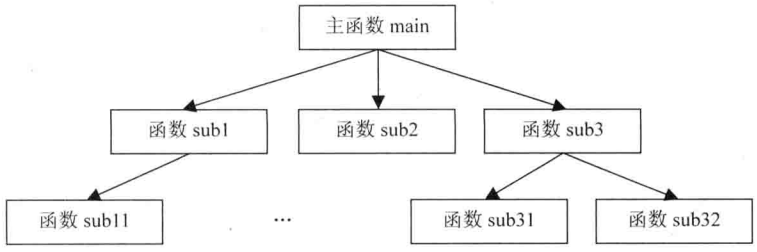

图 1-1　典型的过程式程序结构图

面向过程的程序设计方法具有很多显著的优点，但其缺点也是显而易见的。现实世界中的事物都具有两种属性：静态属性和动态属性。例如，对于一个人来说，姓名、身高、网名等都属于其静态属性，而走路、睡觉、学习等属于动态属性。对于计算机来说，其品牌、CPU 型号、安装的操作系统等属于静态属性，而开机、关机、运行程序等属于动态属性。用计算机解决实际问题时，通常使用一组数据来描述事物的静态属性，而用一组函数来描述事物的动态属性。在面向过程的程序设计中，数据和函数是相互独立的，即程序中的数据和操作它们的方法(函数)相互分离。当需要修改程序中的某些数据描述方法时，使用该数据的所有函数都可能要进行相应的修改。过程化程序设计的这种特点使得程序维护很不方便。在程序设计过程中，当所要解决的问题较为复杂，程序规模较大时，程序员必须细致地考虑程序设计的每个细节，准确考虑程序设计过程中所发生的所有问题。同时，由于各种图形用户界面(Graphics User Interface, GUI)软件的应用，要求应用软件必须随时响应用户的各种操作，因此，软件的功能很难用过程来描述与实现。

为了摆脱软件开发的困境，消除面向过程设计的局限性，在 20 世纪 80 年代出现了面向对象的程序设计(object oriented programming, OOP)。

1.2.2　面向对象的程序设计

面向对象的程序设计方法用更加接近现实问题的对象模型对实际问题进行建模。对象模型将数据和函数封装在一起，数据用来描述实际问题的静态属性，函数用来描述实际问题的动态属性。程序由不同类型的多个对象组成，对象之间通过互发消息进行通信，它们相互配合协作，共同实现程序的功能。

与强调算法的面向过程的程序设计方法不同，面向对象的程序设计强调的是数据。过程化程序设计总是试图使问题满足实际的过程性方法，而对象化的程序设计则是试图让语言来满足实际问题的要求。从软件工程的角度来看，面向对象的程序设计方法具有以下几个显著优点。

(1)更好的模块化。面向对象的程序设计方法的基础就是模块，它通过类和对象将数据和操作这些数据的方法封装在一起构成程序模块。

(2)更高级的抽象。面向对象的程序设计方法不仅支持过程抽象，而且支持数据抽象。面向对象的程序设计方法的类就是一种抽象的数据类型，它描述某一类对象共有的属性，是定义对象的模板，而对象是类的具体实例。此外，某些面向对象的程序设计语言(如 C++)还支持参数化抽象，即通过类模板将类成员的数据类型参数化。高级的抽象性使程序模块的可重

用性更高。

(3)更好的信息隐藏性。在面向对象的程序设计方法中，允许为类的成员设置访问权限，这样就可以将数据隐藏在对象中，用户只能通过类的公有接口来访问类的对象。

(4)低耦合、高内聚性。在面向对象的程序设计方法中，一个对象就是一个独立的单元，其内部各元素用于描述对象的本质属性，它们之间联系紧密，具有高度的内聚性。而不同对象之间只能通过外部接口发送消息来相互通信，如果一类对象的内部结构发生变化，只要它的外部接口保持不变，就不会影响其他的程序模块，所以面向对象的程序模块之间具有更低的耦合性。低耦合性和高内聚性符合软件工程的模块独立原理，是软件重用的基础和保证。

(5)更好的可重用性。继承性和多态性是面向对象的程序设计方法的两个重要属性。通过继承性可以从已有的类型派生出新的类型，新的类型继承了原有类型的属性，并可以增加新属性对原有类型进行扩展。而多态性是指给不同的对象发送相同的消息会导致不同的行为。

面向对象的程序设计方法用类来描述问题，类具有封装性、继承性和多态性。本书从第7章开始详细介绍这一技术。

1.3 C++程序开发

1.3.1 C++程序的开发步骤

高级语言编写的程序从开始编码到运行需要经过以下步骤。

(1)编辑程序。可以在普通的文本编辑器(如 Windows 记事本)或一些专业开发软件(如Microsoft Visual VC++ 6.0)提供的编辑器中对程序进行编码。由高级语言编写的程序称为源程序。C++源程序缺省的扩展名为.cpp，简称 C++程序。

(2)编译。使用编译程序对源程序进行编译，目的是将高级语言编写的源程序翻译成计算机硬件可以识别的二进制机器指令。扩展名为.cpp 的源程序经编译后生成扩展名为.obj 的目标文件。

(3)连接。用连接器将编译成功的目标文件与相应的系统模块连接成扩展名为.exe 的可执行文件。

1.3.2 C++程序的框架

通过一个简单的C++程序可以分析并了解C++程序的基本构成。

【例 1-1】一个简单的C++程序。

```
//*************************
//   一个简单的C++程序
//*************************
#include<iostream.h>
void main( )
{
    cout<<"Hello! "<<endl;
    cout<<"欢迎学习C++!"<<endl;              //A
}
```

程序运行结果

```
Hello!
欢迎学习C++!
Press any key to continue
```

一个完整的C++程序由注释、编译预处理和程序主体构成。

1. 注释

注释是程序员为程序所作的说明，是提高程序可读性的一种手段，一般可将其分为两种：序言注释和注解性注释。前者用于程序开头，说明程序或文件的名称、用途、编写时间、作者以及程序输入/输出说明等；后者用于程序中难懂的地方。为了提高程序的可读性，在书写源程序时，应养成用注释来说明程序功能、语句作用的良好习惯。注释并不是程序的必要部分，和其他高级语言一样，C++编译器在编译一个程序时将跳过注释语句，不对它进行处理。因此，无论源程序中有多少注释，均不会影响程序编译结果。

C++语言提供了两种程序注释方式：一种是由符号“//”开始的注释行，另一种是将介于符号“/*”和“*/”之间的内容作为注释信息。例如：

```
/* ============================
     一个简单的C++程序
============================ */
```

上面的注释信息与下面的注释是等价的：

```
//============================
//     一个简单的C++程序
//============================
```

2. 编译预处理

每个以符号“#”开头的行称为编译预处理行，例1-1中的#include称为文件包含预处理指令。编译预处理是C++组织程序的工具。#include<iostream.h>的作用是在编译之前将文件iostream.h的内容插入程序中。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，并定义了输入/输出流对象cin与cout等。

3. 程序主体

程序中以void main( )开始的部分定义了一个函数，该函数描述了程序的功能。main是函数名。所有的C++程序有且只有一个main函数，通常称该函数为主函数。main函数是整个程序的入口，C++程序总是从主函数的第一条语句开始执行，执行完主函数的所有语句后，程序将结束并返回操作系统。

main前面的void表示该函数没有返回值，即该函数的执行结果不是一个数值。main后的一对圆括号说明main函数运行所需的参数。例1-1中的main函数后是一对空括号，说明本程序运行时无须提供参数。函数所做的操作用相关语句序列实现，必须用花括号{}括起来，称为函数体。

在main函数体中，cout是一个代表标准输出的流设备，它是C++在头文件iostream.h中预定义的对象，前面包含头文件就是为了能在这里使用输出流设备cout。当程序要进行输出时，就需要在程序中指定该对象。输出操作由运算符“<<”来表达，它表示将该运算符右边的数据输出到输出设备(显示器)上。

例 1-1 程序中用双引号括起来的数据“Hello! ”和“欢迎学习 C++!”被称为字符串常量。cout 语句最后的 endl 是系统内部定义的一个换行符，将它输出时其效果是将光标移至下一行，如果有后续输出数据，新的输出将从下一行开始显示。在 C++语言中，一个完整的功能语句均是以分号结束的。

程序执行结束时，在最后增加一行输出“Press any key to continue”，这是系统自动生成的，目的是让用户看清屏幕的输出内容，并提醒用户按任意键后程序将退出并返回原编程环境。

需要特别指出的是，C++程序代码是严格区分大小写的，所以在书写程序时要注意大小写的区别，如函数名 main 不能写成 Main。此外，除了数据，程序中的语法部分不能出现中文字符。例如，例 1-1A 行中的双引号不能是全角中文引号，必须是西文半角形式。

1.3.3 VC++开发环境简介

Microsoft Visual C++ 6.0 集成环境下的 C++程序的开发步骤如下。

(1)在操作系统环境下启动 VC++集成开发环境，打开如图 1-2 所示的界面。

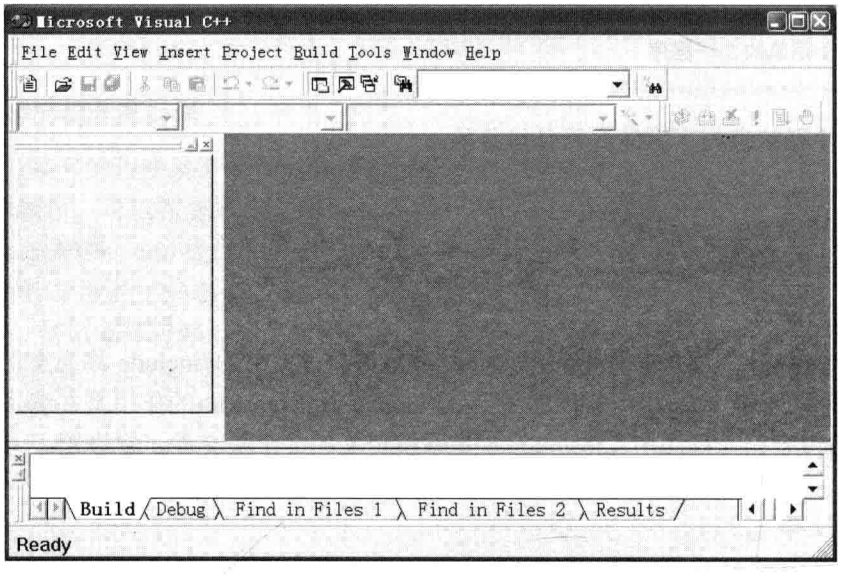

图 1-2 Microsoft Visual C++ 6.0 集成环境界面

(2)选择 Files 菜单下的 New 命令，出现图 1-3 所示的界面(不要直接单击 New 按钮，该按钮用于新建一个文本文件)。该界面缺省标签是为新程序建立工程项目，但对初学者来说，编辑小的源程序不必建立项目，可以直接选择其左上角的 Files 标签，产生图 1-4 所示的界面。

(3)在图 1-4 所示的界面左侧对话框中选中文件类型为 C++Source File(C++源程序文件)，在右边的文本框中填好文件名并选定文件存放目录，然后单击 OK 按钮，打开图 1-5 所示的编程界面，开始输入程序。

(4)输入源程序后，执行 Build 菜单下的 Complie 命令对源程序进行编译，系统将在下方的窗口中显示编译信息。如果无此窗口，可按 Alt+2 组合键或执行 View 菜单下的 Output 命令。

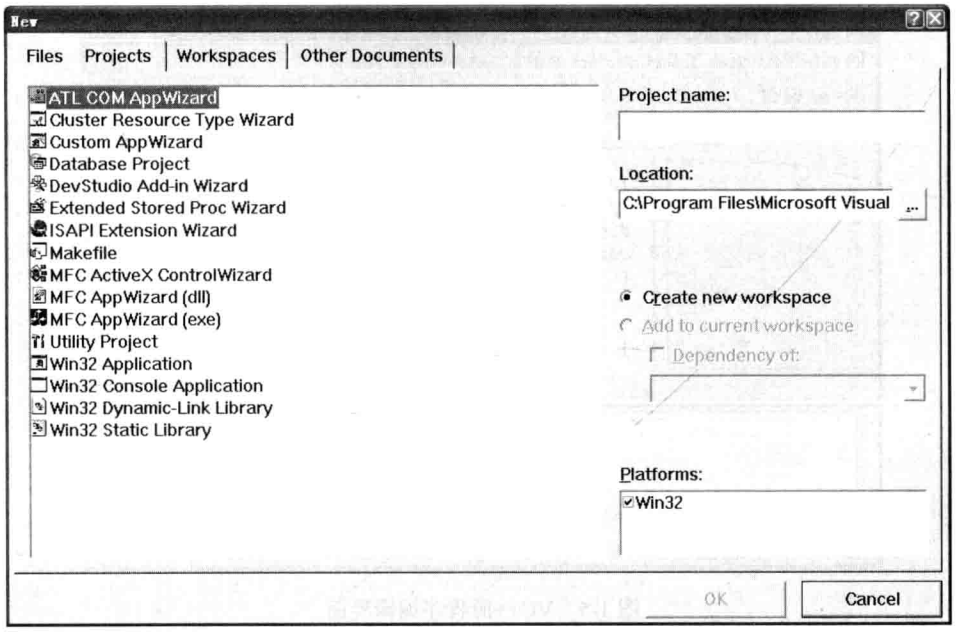

图 1-3　新建 VC++工程项目界面

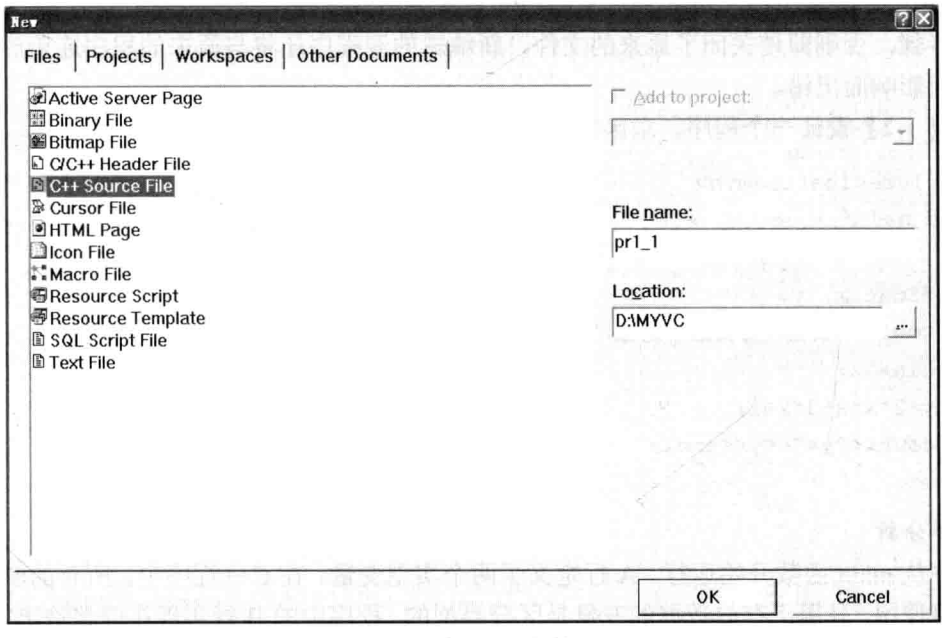

图 1-4　新建 VC++文件界面

如果源程序有语法错误，系统将显示错误所在的行数并给出提示信息，此时可双击相应的错误提示，光标将自动移至错误所在的行，但这仅表示错误表现在这一行，具体错误在哪里可根据系统提示进行判断。一般从第一个错误开始修改，最好是每修改一处错误就编译一次。

如果编译后已无错误提示，则可执行 Build 菜单下的 Build 命令生成相应的可执行文件，随后可执行 Build 菜单下的 Execute 命令运行所编写的程序。

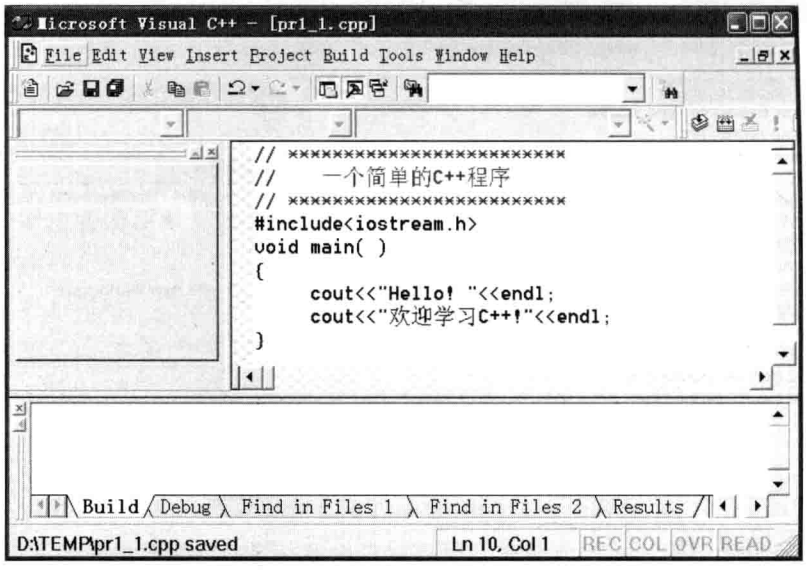

图 1-5 VC++源程序编辑界面

如果要编辑第二个源程序，应该通过 File 菜单下的 Close Workspace 命令来关闭当前工作区(注意：File 菜单下的 Close 命令只能关闭正在编辑的文件，而不能关闭当前工作区)，重复以上步骤，否则即使关闭了原来的文件，新编辑的源程序还将与原来的程序连接成一个文件，相互影响而出错。

【例 1-2】设计一个程序，求函数 $y=2x^2+3x+1$ 的值。

```
#include<iostream.h>
void main( )
{
    float x, y;                          //A
    cout<<"请输入自变量 x 的值:";          //B
    cin>>x;                              //C
    y=2*x*x+3*x+1;                       //D
    cout<<"y="<<y<<endl;
}
```

程序分析

程序从 main 函数开始运行。A 行定义了两个实型变量。在 C++程序中，所有的变量必须先定义再使用，且用于存放数据的变量是区分类型的。程序中的 B 行实现在屏幕输出字符串，提示用户输入数据。C 行的 cin 是输入流设备，其后的“>>”是输入运算符，用于从键盘接收数据并存入变量 x。D 行根据表达式计算变量 y，即函数的值。

程序运行结果

```
请输入自变量 x 的值:0.25↙(↙表示回车键)
y=1.875
```

程序执行 C 行时，将等待用户从键盘输入数据，并按回车键后，程序继续执行。

习 题

1. 简述C++程序的基本组成。
2. C++程序的注释方式有哪几种？
3. C++程序从开始设计到最终看到结果要经过哪几个步骤？
4. 参照本章例题设计一个简单的C++程序，用来计算从键盘输入的两个实数之和。

第 2 章　数据类型与表达式

数据是计算机程序的重要组成部分，每种程序设计语言都有其规定的数据类型，数据类型决定了数据的存储以及数据的操作方式。C++语言的数据类型大致可分为基本数据类型与构造数据类型，本章主要介绍基本数据类型。

2.1　标　识　符

在程序设计中，经常需要用标识符来代表数据、变量等。合法的标识符是指程序中允许使用的构成成分的命名，它们由字符组成。

1. 字符集

C++语言的基本元素均由字符集中的字符组成，包括字母、数字以及键盘上除@、`、$外的其余可显示字符。

2. 关键字

关键字也称保留字，是程序设计语言中约定已具有某种特定含义的标识符，不可以再作其他用途。表 2-1 中列出了 C++语言中常用的关键字。

表 2-1　C++语言中常用关键字

int	double	if	for
char	float	else	while
void	const	switch	do
long	short	break	return
this	struct	continue	private
inline	case	union	protected
operator	default	enum	public
virtual	auto	class	friend
static	extern	signed	delete
register	typedef	unsigned	new

3. 自定义标识符

自定义标识符是指程序中由用户命名的变量名、函数名、类型名等，它由英文字母、数字、下划线组成，不能以数字开头。

用户自定义标识符时需要注意以下几点。

(1) 自定义标识符应尽量做到见名思义。如用 name 表示姓名，用 year 表示年份等。

(2) 不可以将关键字作为自定义标识符。

(3) C++语言中严格区分大小写。如 myName 和 myname 是两个不同的标识符。

【例 2-1】判断下列各符号是否为合法的自定义标识符。

Student_001，static，length，5name，n-Computer，While，_123，s3，&address

合法的自定义标识符有 Student_001，length，While，_123，s3。而 static 是关键字，5name 以数字开头，n-Computer 中含符号“-”，&address 中含符号“&”，故均为不合法的自定义标识符。

2.2　基本数据类型

基本数据类型是指 C++语言中已预定义的、不可再进一步分割的数据类型，构造数据类型是指由一种或几种数据类型组合而成的数据类型，每种类型数据都有常量与变量之分。各种构造数据类型将在后面章节介绍。

2.2.1　基本数据类型简介

C++语言的基本数据类型包括 void(空类型)、bool(布尔型)、int(整型)、char(字符型)、float(单精度)、double(双精度)等。不同数据类型的数据所占内存空间的大小不同，其所能表示数值的取值范围也不同。表 2-2 是对 C++语言中常用的基本数据类型的描述。

表 2-2　C++语言中常用的基本数据类型

名称		类型	长度/B	取值范围
布尔型		bool	1	true 或 false
字符型		char	1	–128~127
整型		int	4	-2^{31}~$(2^{31}-1)$
实型	单精度	float	4	-10^{38}~10^{38}
	双精度	double	8	-10^{308}~10^{308}
空类型		void	0	无值

为了更准确地描述数据类型，C++语言中还提供了 4 个关键字：short、long、unsigned 和 signed，用来修饰整型数据。另外，可以用关键字 unsigned 和 signed 修饰字符型数据。相应的基本数据类型如表 2-3 所示。

表 2-3　C++语言中经修饰的基本数据类型

名称	类型	长度/B	取值范围
无符号字符型	unsigned char	1	0~255
短整型	short int	2	–32768~32737
长整型	long int	4	-2^{31}~$(2^{31}-1)$
无符号短整型	unsigned short int	4	0~65535
无符号长整型	unsigned long int	4	0~$2^{32}-1$