普通高等教育“十二五”规划教材

C语言程序设计

主　编　吴登峰　邢鹏飞
副主编　耿　姣　李　婧　宁海涛

中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十二五”规划教材

C语言程序设计

主　编　吴登峰　邢鹏飞

副主编　耿　姣　李　婧　宁海涛

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书主要讲解C语言的常用语法规则、使用特点、程序设计的基本思路及实现方法，并强调模块化的程序设计理念。

本书共9章，包括C语言概述、C语言基础、选择结构程序设计、循环结构程序设计、数组、函数、指针、构造数据类型、文件等内容。为了使读者更好地掌握各章节内容，每章配有精选习题。本书还增加了对C语言部分新内容的讲解，使程序更加规范。

本书章节安排合理、基本概念清晰、讲解深入浅出、文字流畅、通俗易懂，适合初学者学习。本书主要面向高等院校理工类专业学生，可以作为高等院校各专业的正式教材，也可以用作自学教材。

本书提供程序源代码，读者可以从中国水利水电出版社网站和万水书苑上下载，网址为：http://www.waterpub.com.cn/softdown/和 http://www.wsbookshow.com。

图书在版编目（CIP）数据

C语言程序设计 / 吴登峰，邢鹏飞主编. -- 北京 : 中国水利水电出版社，2015.1
普通高等教育“十二五”规划教材
ISBN 978-7-5170-2776-8

Ⅰ. ①C… Ⅱ. ①吴… ②邢… Ⅲ. ①C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(2014)第308665号

策划编辑：石永峰　　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语言程序设计
作　者	主　编　吴登峰　邢鹏飞 副主编　耿　姣　李　婧　宁海涛
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　14印张　345千字
版　次	2015年1月第1版　2015年1月第1次印刷
印　数	0001—2000册
定　价	28.00元

前　言

C 语言是应用最广泛的程序设计语言之一，它同时具备高级语言的特点和汇编语言的部分功能，功能强大、语句简洁、编译高效、结构清晰、移植性好，既能适合编写应用软件，也能用于开发系统软件。因此，C 语言成为高校程序设计课程的首选语言。

本书系统全面地介绍 C 语言的基本概念、语法结构，以及结构化程序设计的思想和方法。从知识点到程序设计实例，由浅入深、循序渐进，使读者理解并掌握相关知识，进一步培养编程能力。

全书共 9 章，具体内容如下：

第 1 章，主要介绍 C 语言的发展历程和基本特点，C 语言程序的基本结构，C 语言程序的开发环境和运行方式。

第 2 章，主要讲解 C 语言的基本语法，包括计算机中数据的表现形式、基本数据类型、运算符和表达式、基本语句和常用的输入输出函数，并通过 C 语言基本知识结构完成顺序结构程序设计。

第 3 章，主要讲解选择结构程序设计，包括关系运算符和表达式、逻辑运算符和表达式、if 语句和 switch 语句的用法，并用其实现选择结构程序设计。

第 4 章，主要讲解循环结构程序设计，包括 for 循环结构、while 循环结构和 do…while 循环结构，并实现相应的循环结构程序设计。

第 5 章，主要讲解数组，包括一维数组、二维数组和字符数组的应用。

第 6 章，主要讲解函数及模块化程序设计，包括变量的作用域与存储方式、函数的定义及基本调用、函数的嵌套调用及递归调用、C 语言的预处理器及库函数、模块化程序设计思想。

第 7 章，主要讲解指针，包括指针的基本概念和用法、指针与一维数组、二维数组、字符数组、函数的结合使用，进一步讲解指针数组和多级指针以及动态存储空间的分配与释放。

第 8 章，主要讲解构造数据类型，包括结构体、链表、共用体、枚举等构造数据类型的用法，在思想上逐渐与面向对象程序设计接轨。

第 9 章，主要讲解文件，包括文件及文件指针的基本概念，文件的常用操作。

C 语言程序设计是一门实践性很强的课程，学习者必须经过大量的编程训练，才能逐步理解并掌握程序设计的方法。本书根据每章知识点和全国计算机等级考试二级 C 语言考试大纲编写了课后习题。希望读者尽量完成课后习题，以巩固所学知识。

本书适用于普通高等院校计算机专业本科、理工类各专业本专科学生使用，也可以作为全国计算机等级考试的参考用书。

本书由吴登峰、邢鹏飞担任主编，耿姣、李婧、宁海涛担任副主编，参与本书编写工作的还有张天、陈然、刘妍、杨鑫、燕杨。

由于作者水平有限，书中难免存在谬误之处，恳请读者批评指正。

编　者

2014 年 10 月

目　　录

第 1 章　C 语言概述

欢迎来到 C 语言的世界，C 语言是国际上广泛流行的专业化编程语言，深受软件编程人员的欢迎，现在，我们将一起学习这一强大而实用的计算机编程语言。

1.1　C 语言的生命力

1.1.1　C 语言的发展历程和趋势

C 语言是在 BCPL（Basic Combined Programming Language）的基础上发展而来的，BCPL 的原型是 ALGOL 60（也称 A 语言）。ALGOL 60 是计算机发展史上的首批高级语言，更适合于数值计算。1963 年，剑桥大学将 ALGOL 60 发展成为 CPL（Combined Programming Language）。CPL 比 ALGOL 60 更加接近硬件，但规模比较大，实现困难。1967 年，剑桥大学的 Matin Richards 对 CPL 进行了简化，形成了 BCPL。1970 年，美国贝尔实验室的 Ken Thompson 在 BCPL 的基础上做了进一步的简化，设计出了简单且接近硬件的 B 语言（取 BCPL 的第一个字母），并且采用该语言编写了第一个 UNIX 操作系统，并在 PDP-7 上实现。1973 年，贝尔实验室的 Dennis. M. Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言既保持了 BCPL 和 B 语言的精练、接近硬件的优点，又克服了它们过于简单、无数据类型的缺点。开发 C 语言的目的在于尽可能降低它所写的软件对硬件平台的依赖程度，使之具有可移植性。同年，Ken Thompson 和 Dennis. M. Ritchie 合作把 UNIX 系统的 90%以上的代码用 C 语言重新编写，完成 UNIX 第 5 版。1977 年，D.M. Ritchie 发表了不依赖具体机器的 C 语言编译文本——可移植的 C 语言编译程序，简化了 C 语言移植到其他机器上的工作，推动了 UNIX 操作系统在各种机器上的实现。随着 UNIX 的广泛使用，C 语言先后移植到大、中、小型机上，得以迅速推广，很快风靡全球，成为世界上应用最广泛的高级语言。

1978 年，Brian W. Kernighan 和 Dennis. M. Ritchie 联合撰写了影响深远的名著《The C Programming Language》，成为第一个事实上的 C 语言标准。1983 年，美国国家标准化协会（ANSI）成立了专门的委员会，根据当时存在的不同 C 语言版本进行改进和扩充，指定了 C 语言标准草案——83 ANSI C。1987 年 ANSI 又公布了新的标准——87 ANSI C。1989 年，ANSI 公布了更加完整的 C 语言标准——ANSI X3（也称 ANSI C 或 C89）。1990 年，国际标准化组织（International Standard Organization，ISO）接受 C89 为 ISO C 的国际标准（ISO/IEC9899:1990）。1994 年和 1995 年，ISO 又先后修订了 C 语言标准，称为 1995 基准增补 1（ISO/IEC/9899/AMD1:1995）。1999 年，ISO 又对 C 语言标准进行修订，在原有基础上，增加了 C++中的一些功能，命名为 ISO/IEC9899:1999。2011 年 12 月 8 日，ISO 正式发布了 C 语言的新标准 C11，提高了对 C++的兼容性，并加入对多线程的支持等功能，命名为 ISO/IEC 9899:2011。

但在实际的使用中，目前不同的公司对 C 语言编译系统的开发，并未完全实现最新的 C 语言标准，它们多以 C89 为基础开发。而初学者所用到的编程知识都包含在 C89 范围内。本

书中的示例程序都可以在目前主流编译系统（Visual C++ 6.0、Turbo C++3.0、GCC）上编译和运行。

1.1.2 C语言的特点

随着计算机技术的不断发展，20世纪90年代出现了面向对象的编程语言和开发平台，许多软件开发商开始转向使用C++和Java语言来进行大规模的项目开发。不管这些较新的语言如何流行，C语言在软件开发产业中仍然是一种重要的编程语言，已被广泛用于嵌入式系统编程，并在汽车、照相机、DVD播放器等现代化设备的微处理编程中起着举足轻重的作用。此外，C语言还适用于操作系统的开发，伴随UNIX和Linux的成长，它将扮演更重要的角色。因此，在未来很长的一段时间内，C语言仍将保持强劲的势头。

C语言与其他编程语言相比，有着自己独特的特点：

（1）语言简洁、灵活方便

C语言共有32个关键字，9种控制语句，程序书写形式自由灵活。C语言程序通常由多个函数组成，便于模块化和结构化编程，使得编写的程序结构清晰明了、可读性强。

C是一个很小的内核语言，只包含极少的与硬件有关的部分，有关输入、输出、文件操作的语句和动态内存管理的语句不是由C语言本身提供，而是借用编译系统提供的库函数来实现。

（2）表达能力强

C语言不仅提供了丰富的运算符和数据类型，还提供了强大的功能库。使得程序员可以快速、灵活地编写程序，精确地控制计算机按照自己的意愿工作。

（3）高效率的编译性语言

C语言生成的目标代码质量高，运行速度快。对于较大的程序，源代码可以分别存放，单独编译后再链接在一起，形成可执行文件。

（4）可移植性好

采用C语言编写的程序基本上可以不做修改，直接运行于各种型号的计算机和各种操作系统。

（5）运算符和数据类型丰富

C语言包含34种运算符，运算符种类丰富，表达式类型多样，使用灵活。

C语言提供了整型、浮点型、字符型、数组类型、指针类型、结构体类型和共用体类型等基本类型，C99增加了超长整型（long long）、布尔类型（bool）和复数浮点类型（float _Complex）等，使C语言能适用更多的环境。

（6）语法限制少，设计自由度大

C语言允许程序员有更大的自由度，编译时放宽了语法检查，例如，对数组下标越界不进行检查，整型量、字符型量与逻辑型量可以通用等。由于语法检查比较松，对程序员的要求会更高，特别是初学者不易掌握。

（7）允许直接访问物理地址

C语言既具有高级语言的功能，又具备低级语言的很多功能，使它既是通用的程序设计语言，又是系统描述语言。C语言允许直接访问物理地址，还能够进行位运算，实现汇编语言的大部分功能，可以直接对硬件进行操作。

1.2　C 语言程序示例

1.2.1　C 语言程序示例

C 语言程序到底是什么样子的？让我们一起来看一个简单的 C 程序（例 1.1），并尝试读懂该程序所做的事情。

例 1.1　在屏幕上输出如下信息“This is the first program.”。

程序清单如下：（文件名 1_1.c）

```
#include <stdio.h>                          /*编译预处理指令*/
int main()                                  //定义主函数，C 程序的开始
{                                           /*函数的开始标志*/
    printf("This is the first program.\n"); /*输出指定信息*/
    return 0;                               /*函数正常结束返回值为 0*/
}                                           /*函数的结束标志*/
```

例 1.1 运行结果如图 1.1 所示。其中第 1 行“This is the first program.”为程序运行后的结果。第 2 行为 Visual C++ 6.0 编译系统在输出运行结果后自动输出的信息，即“按任意键继续”，按键盘上的任意按键后，运行结果窗口消失。

图 1.1　例 1.1 的运行结果

下面分析这个最简单的 C 语言程序的结构：

1. 文件包含预处理指令

例 1.1 的第 1 行“#include <stdio.h>”是一个 C 预处理指令，该行告诉编译器包含头文件 stdio.h 中的全部信息，其作用相当于在程序中该行所在的位置键入了 stdio.h 的完整内容。stdio.h 文件是标准输入输出头文件（standard input/output header），由 C 编译系统提供，包含了有关输入和输出的函数（如 printf()、scanf()等）的信息以供编译器使用。在 C 语言中，将出现在文件顶部的信息集合称为头（header），这些文件通常以.h 作为扩展名。当然文件包含预处理指令也可以包含用户定义的其他文件，它的基本格式为：

#include <文件名>　或者　#include "文件名"

这两者之间的主要区别是：使用<>时，编译系统到存放 C 库函数头文件的目录中去寻找要包含的文件，而使用""时系统先在用户当前目录中去寻找要包含的文件，若找不到再按照<>的方式进行查找。

文件包含预处理指令的作用

“文件包含预处理”指令非常有用，它可以节省程序设计人员的重复劳动。例如，某个单位在进行某个项目开发的时候需要使用一组固定的数据（如重力加速度为 9.8、圆周率为 3.1415926、…），项目组可以把这些经常用到的固定的数据定义成一个头文件，项目组的成员需要使用这些数据时只需使用#include 命令将头文件包含在自己的文件里就可以了，这样项目组中的每个成员就不必重复定义这些数据，而且也便于修改相应的值。

2. main()函数

例 1.1 的第 2 行“int main()”代码开始定义 main()函数。C 程序中必须有一个 main()函数，它表示 C 程序中的主函数，C 程序的执行总是从该函数开始。如果在 main()函数中调用了其他函数，调用结束后流程将返回到 main()函数，在 main()函数中结束整个程序的运行。

int 指明 main()函数的返回类型，由于 int 表示整型，意味着 main()函数的返回类型是整型。（C99 建议把 main 函数的返回值类型指定为 int 型，要求函数返回一个整数值。在 C99 以前 main 函数的返回值类型经常使用 int 型。）

例 1.1 的第 5 行“return 0;”的功能是当 main 函数正常运行结束时，给调用 main 函数的操作系统返回函数值 0。

3. 大括号与函数体

例 1.1 的第 3 行和第 6 行为一对大括号“{...}”，这对大括号属于 main 函数，它们划定了 main 函数的界限。在 C 语言中，所有的函数都使用花括号来表示函数的开始和结束，在大括号之间的部分，就是函数体。函数体由若干语句构成，这些语句描述了函数的功能。

在 C 语言中，大括号还有一个功能，就是用来把某些语句聚集成一个单元或代码块，这些单元或代码块称作复合语句。

4. 输出函数

例 1.1 的第 4 行“printf("This is the first program.\n");”代码完成在屏幕上输出一句话“This is the first program.”，这个功能由函数 printf()实现。printf 是函数的名称，是 C 语言编译系统提供的函数库中的输出函数，函数中的双引号内部分称为字符串，即将“This is the first program.”原样输出，“\n”是换行符，表示在输出“This is the first program.”后，使显示屏上的光标转到下一行的开头。

在每个语句的结尾都有分号，表示语句结束。

5. 注释

在例 1.1 中，还有诸如“/*文件包含预处理*/”和“//定义主函数，C 程序的开始”这样的内容。C 语言使用“//”和“/* */”在程序中间添加注释内容，帮助程序员理解程序的功能。其中，“//”表示从符号“//”开始到本行结束的内容是注释内容，注释内容只能在一行之内；“/* */”表示符号“/*”和符号“*/”中间的内容是注释内容，注释内容允许包含多行内容。

```
/*注释的第一行，
  注释的第二行，
  这一行仍然是注释。*/
//这一行内容是注释。
```

注释内容可以使用汉字、英语字符和各种符号，C 语言编译系统不检查，也不编译注释内容。

随堂练习

仿照例 1.1，请编写一个程序输出“My name is XXX.”。

例 1.2 求两个整数的和。

解题思路：设置两个变量 a 和 b 存储两个整数，使用 C 语言加法运算求和，并将和使用赋值运算存储在另一个变量 sum 中，最后输出 sum 的值。

程序清单如下：（文件名 1_2.c）

```
#include <stdio.h>                          //编译预处理指令
int main()                                  //定义主函数，C 程序的开始
{                                           //函数的开始标志
    int a, b, sum;                          //声明 a、b、sum 均为整型变量
    a = 222;                                //将整数 222 放在变量 a 中存储
    b = 333;                                //将整数 333 放在变量 b 中存储
    sum = a + b;                            //将整数 a 和 b 的和放在变量 sum 中存储
    printf("sum is %d\n",sum);              //输出 sum 的值
    return 0;                               //函数正常结束返回值为 0
}                                           //函数的结束标志
```

例 1.2 运行结果如图 1.2 所示。

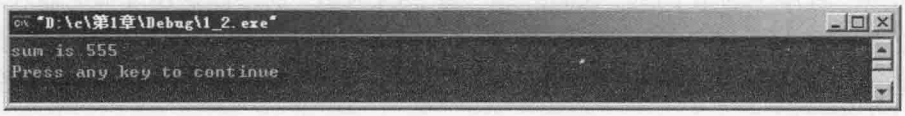

图 1.2　例 1.2 的运行结果

例 1.2 的功能是求两个整数的和，在例 1.1 的基础上增加了变量的使用。第 4 行为变量的声明，其中，“int”是 C 语言的关键字，含义是整型，功能是定义后面的变量为整型变量，即第 4 行定义 a、b 和 sum 为整型变量。第 5 行和第 6 行中，“=”是 C 语言的运算符号，含义是赋值，功能是将“=”右侧的表达式的结果存储在“=”左侧的变量中，即第 5 行和第 6 行是将 222 和 333 两个整数分别存储在变量 a 和 b 中。第 7 行，首先计算变量 a 和 b 的和，也就是 222 和 333 两个整数的和，结果存储在变量 sum 中。第 8 行，输出运行结果，函数 printf 的括号中包含两部分内容，第一部分是“sum is %d\n”，其中，“sum is”是按照用户需要原样输出的信息（与例 1.1 相同），“%d”是将变量使用“十进制整数”的形式输出，“\n”是换行符（与例 1.1 相同）；第二部分是要输出的变量 sum。在执行函数 printf 时，将变量 sum 中存储的值取代“%d”输出。

在例 1.2 的运行结果中，程序的输出结果为“sum is 555”，然后换行。“Press any key to continue”为编译系统给出，与程序无关。

随堂练习

1. 将例 1.2 中的“\n”去掉，观察程序运行结果，并思考原因。
2. 仿照例 1.2，请编写一个程序求 3 个整数的和。

例 1.3　求两个整数的和。

程序清单如下：（文件名 1_3.c）

```
#include <stdio.h>
int add(int x, int y)                       //定义函数 add，add 为函数名，x、y 为形式参数
{
    return (x + y);                         //将 x + y 的和返回，通过 add 带回到调用函数的位置
}
int main()
```

```
{
    int a, b, sum;                                  //定义变量 a、b 和 sum
    printf("Please input two number(like:3,5): \n");  //使用 printf()函数打印输入提示
    scanf("%d,%d", &a, &b);                         //使用 scanf()函数对变量 a，b 赋值
    sum = add(a, b);                                //调用函数 add，并将函数返回值赋值给 sum
    printf("%d + %d = %d\n", a, b, sum);            //使用 printf()函数输出结果
    return 0;
}
```

程序的第 2 至 5 行为一个函数，函数名称为 add，功能为求两个给定的整数的和。

运行程序后，出现如图 1.3 所示窗口，其中，第 1 行内容“Please input two number(like:3,5):”为程序第 9 行 printf 函数输出的结果，而且由于第 9 行 printf 函数中有“\n”，所以闪动的光标出现在结果窗口的第 2 行。结果窗口并未出现“Press any key to continue”的提示，说明程序并未运行结束。程序中的第 10 行“scanf("%d,%d", &a, &b);”的功能是在屏幕上输入两个整数，如“3,5”，其中双引号中的两个“%d”要求输入两个整数，两个“%d”中间有逗号说明输入两个整数时要用逗号分隔，后边的“&a, &b”是将输入的两个整数按照顺序保存在变量 a 和 b 中（如图 1.4 所示）。

图 1.3　程序 1.3 的运行结果 1

图 1.4　程序 1.3 的运行结果 2

输入两个整数后，单击回车，出现如图 1.5 所示窗口。程序的第 11 行“sum = add(a, b);”中，“add(a, b)”是调用第 2 至 5 行的 add 函数，功能为求变量 a 和 b 的和，本例就是整数 3 与 5 的和，即“add(a, b)”的结果为 8，然后将 8 赋值给变量 sum。程序的第 12 行“printf("%d + %d = %d\n", a, b, sum);”双引号中的三个“%d”表示后面的三个变量 a、b 和 sum 以整型数据的格式输出，在结果窗口中表现为“3 + 5 = 8”。

图 1.5　程序 1.3 的运行结果 3

现在让我们一起分析程序 1.3，以便对 C 程序有一个初步的了解。

图 1.6 总结了一个典型的 C 程序的各个组成部分，从中可以看到 C 程序是由函数构成的，一个 C 源程序有且仅有一个 main()函数，但是可以包含多个其他函数，当然也可以没有其他函数，仅有一个 main()函数。

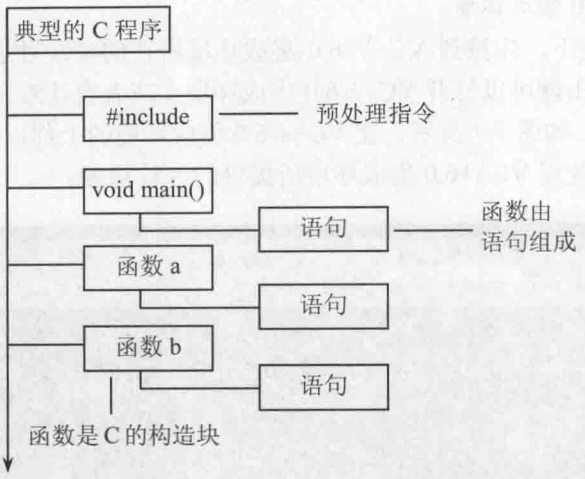

图 1.6　C 程序剖析图

例 1.3 是由一个 main()函数和一个 add()函数组成的 C 程序。在例 1.1 和例 1.2 的基础上，增加了函数的使用。

一个函数通常由两部分组成：函数的首部和函数体。函数首部包括函数名、函数类型、函数参数（形参）、参数类型。

例如：例 1.3 中 add 函数的首部如下

```
int      add      (int          x ,          int          y)
 ↓        ↓        ↓            ↓            ↓            ↓
函数类型  函数名  函数参数类型  函数参数名  函数参数类型  函数参数名
```

一个函数名后面必须跟一对圆括号，函数参数可以有多个，也可以没有。

函数体是紧跟函数首部下面的大括号内的部分，包括了实现函数特定功能的变量定义及若干执行语句。有时，函数体也可以没有变量定义和执行语句，只有一对大括号，就是对被调函数 add 的一个声明。

对函数的声明需与函数定义的函数名，形式参数的个数、类型和顺序都保持一致。当然，在函数声明的语句中，也可省略具体的形式参数名字，只给出形式参数的类型。如：

```
int add(int, int);
```

如果被调函数的定义出现在主调函数之前，则可以不加声明。因为编译系统已经知道了被调函数，并根据函数定义的相关信息对函数的调用作出正确性检查。

1.2.2　运行 C 程序的方法

C 程序在什么环境中编写，如何运行，怎样查看结果呢？在这一小节里，我们将主要介绍这些知识。

C 程序（*.c）编写好后，必须先对其进行编译得到目标程序文件（*.obj），再将目标程序与系统提供的库函数等进行连接，才能得到可执行的目标程序（*.exe）。为了编译、连接和运行 C 程序，必须要有相应的 C 编译系统。目前使用的 C 编译系统大多都是集成环境（IDE）的，把程序的编辑、编译、连接和运行等操作全部集中在一个界面上进行。本书将以 VC++ 6.0 集成开发环境作为工具来进行 C 程序开发。

1. 进入 VC++ 6.0 集成环境

在 Windows 环境下，先找到 VC++ 6.0 集成环境所在的安装目录，从中找到可执行文件 MSDEV.exe，直接双击就可以打开 VC++ 6.0 集成环境（或者直接双击电脑“所有程序”中的 VC++ 6.0 快捷方式），如图 1.7 所示。在 VC++ 6.0 集成环境的上部，有一行“主菜单”，我们可以通过这些菜单来使用 VC++6.0 集成环境所提供的各种功能。

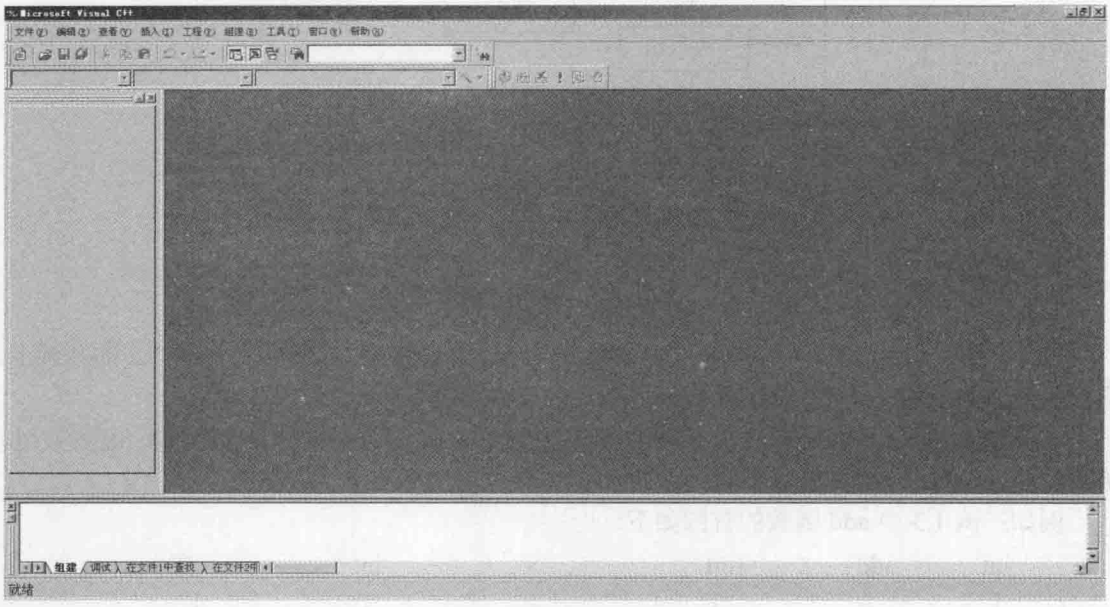

图 1.7 VC++ 6.0 集成环境

2. 新建源文件

下面建立新的 C 程序文件。在“文件”菜单下选择“新建”，弹出如图 1.8 所示对话框，选择“文件”选项卡，单击选择“C++ Source File”，在“文件名”文本框中输入要建立的 C 语言源程序名字，如 1_4.c，在“位置”下方路径右侧，点击...按钮，选择相应的保存位置，点击“确定”按钮，完成源文件建立。进入编辑界面，在右侧的空白处就可以编辑 C 语言源程序了，如图 1.9 所示。

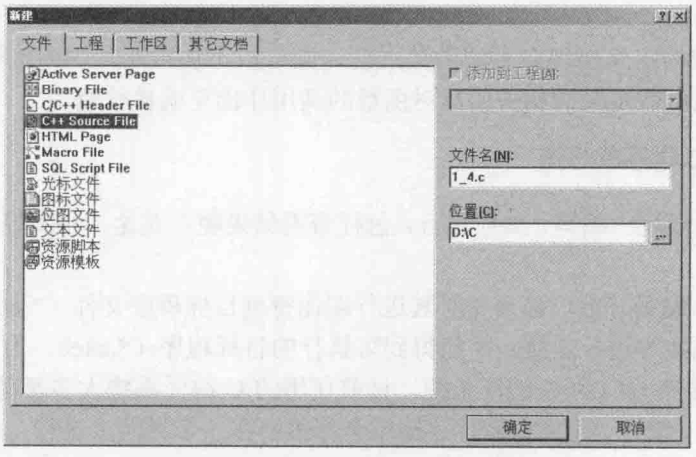

图 1.8 “新建”源文件对话框

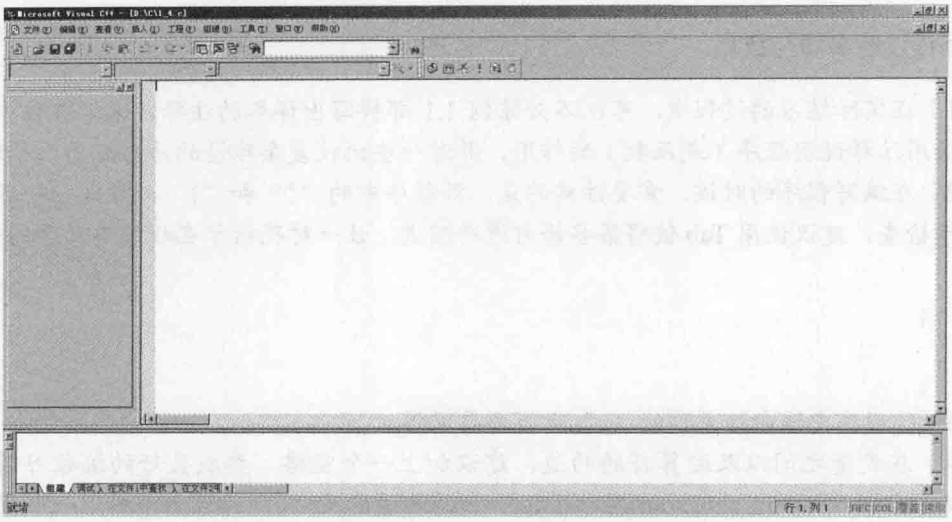

图 1.9　源程序编辑界面

3. 编译源程序

编译源程序的时候，选择“组建”菜单，在其下拉菜单中选择“编译”（也可以使用组合键 Ctrl + F7）。这时，在集成环境的下方消息（Build）框中会出现编译情况说明，若说明为 0 个错误（error），0 个警告（warning），则表明程序没有语法错误，可以继续连接目标程序；若存在错误（error）不为 0，则说明程序存在错误需要修改，此时无法进行后续的连接目标源文件。

4. 连接目标程序

源程序编译没有错误后，就生成目标程序文件，此时还需要与系统提供的库函数等进行连接，才能得到可执行的目标程序文件。连接的方法是选择菜单“组建”，在其下拉菜单中选择“组建”，如果不出现错误，就会得到一个后缀为.exe 的可执行文件。

5. 执行程序

如果连接没有错误，则可以在集成环境中执行程序，并查看运行结果。选择菜单“组建”，在其下拉菜单中选择“执行”（也可以使用 Ctrl + F5 组合键），系统就会执行已经编译和连接好的可执行文件。运行时，如果程序需要输入数据，则屏幕会切换到运行窗口，等待输入数据。在输入后，就会显示运行结果。

6. 退出 VC++ 6.0 集成环境

程序调试完成后，可以选择“文件”菜单中的“退出”或者点击“关闭”按钮来退出 VC++ 6.0 集成环境。

1.3　小结

本章概述了 C 语言的发展历程、C 语言的发展趋势和 C 语言的特点，并通过具体的示例程序对 C 程序中的文件包含预处理指令、main()函数、C 语言的注释、变量的定义、输入输出函数的使用、函数的定义与调用、赋值运算符的使用以及怎样使用 VC++ 6.0 集成环境来运行一个 C 程序进行了介绍。

学习这一章，需要把注意力放在怎样编写第一个程序上面，通过本章的学习期望能够仿照例 1.1 编写一些简单的程序，并能够对里面的每个知识点的使用方法熟练掌握。

编程注意事项与技巧

（1）在实际练习的过程中，可以不必像例 1.1 那样写出详尽的注释，通常在程序（或函数）开头用注释说明程序（或函数）的作用，并对一些比较复杂难懂的语句给出注释即可。

（2）在编写程序的时候，需要注意的是，源程序中的“{”和“}”必须是一一匹配的。为了便于检查，建议使用 Tab 键将某些语句进行缩进，让一对花括号在程序中处于同一列中。如：

```
{
    …
}
```

（3）C 程序中语句结束时的“;”千万不要漏掉。

（4）在变量之间以及运算符的两边，建议加上一个空格，养成良好的编程习惯。如：printf("%d + %d = %d\n", a, b, sum)在“+”、“=”的两边以及“,”的后面都加了一个空格。

（5）使用 VC++ 6.0 集成环境编译 C 程序出错时，光标所停留的那一行有可能不是出错行，需要在这一行的上下几行中查找错误。

1.4 习题

1. 请编写一个 C 程序，输出以下信息：

```
**************************************
*          Hello, C program!         *
**************************************
```

2. 请编写一个 C 程序，从键盘上输入两个整数，求出这两个数的积，并将结果输出在屏幕上（提示：C 语言中的乘法运算符是“*”，这里不考虑两个较大的数相乘的情况）。