高等学校计算机应用规划教材

C语言程序设计

- Visual C++ 6.0的安装及使用
- 数据类型及基本运算量
- 顺序结构程序设计
- 选择结构程序设计
- 循环结构程序设计
- 数组
- 用户自定义函数
- 预处理命令
- 用户自定义数据类型
- 指针
- 文件
- 位运算

梁海英 主编

李淑梅 白文秀 侯锟 副主编

清华大学出版社

高等学校计算机应用规划教材

C 语言程序设计

梁海英　主　编

李淑梅　白文秀　侯　锟　副主编

清华大学出版社

北　京

内 容 简 介

本书按照程序设计的体系结构，系统地介绍了 C 程序设计的基本思想及基本方法。全书内容分为两部分。第一部分(第 1～7 章)依次介绍了 C 程序设计的基本应用：包括 C 程序的基本结构、基本运算量，以及结构化程序设计的顺序结构、选择结构、循环结构、数组及函数的开发方法与应用实现；第二部分(第 8～12 章)详细介绍了 C 程序设计的高级应用：包括预处理、结构体、共用体、指针、文件及位运算等应用。

本书从实用的角度出发，内容选取先进精准、组织循序渐进、讲解文字精练；各部分辅助图表、结合实例、深入浅出、结构清晰；典型实例精挑细选、算法分析流程图化、程序结构错落有致、程序结果真实有效；各章习题针对性强、题型丰富；免费提供电子课件、源文件及习题答案；详细介绍了开发环境 Visual C++ 6.0 的使用方法，全部例题均在此环境中成功运行。

本书可作为高等学校非计算机专业本科生的计算机通识教材，也可作为计算机相关专业的程序设计入门教材、计算机技术的培训教材，还可作为全国计算机等级考试的参考用书和编程爱好者自学 C 语言的自学教材。

本书对应的电子教案、习题答案和实例源文件可以到 http://www.tupwk.com.cn/downpage 网站下载。

图书在版编目(CIP)数据

C 语言程序设计/梁海英 主编. —北京：清华大学出版社，2013.2

(高等学校计算机应用规划教材)

ISBN 978-7-302-31359-5

Ⅰ. ①C… Ⅱ. ①梁… 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(2013)第 013979 号

责任编辑：胡辰浩　袁建华
封面设计：牛艳敏
责任校对：成凤进
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62796045

印 装 者：三河市李旗庄少明印装厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：18.25　　**字　　数**：421 千字
版　　次：2013 年 2 月第 1 版　　**印　　次**：2013 年 2 月第1 次印刷
印　　数：1～4000
定　　价：32.00 元

产品编号：051094-01

前　言

C 语言程序设计简单易学，具有广泛的用途，是非常适合高等学校各专业学生学习的程序设计语言，又是计算机学科的程序设计基础课。通过 C 语言程序的学习，读者可以运用相关知识和技能更好地进行算法和程序的设计，也为后继课程的学习打下良好的基础。

我们基于多年的丰富教学经验及素材积累，精心编写此书，目的是让初学者能循序渐进地掌握程序设计的思想，系统地掌握 C 语言程序设计的方法，从实用的角度出发，选取适当的相关案例，配备精练的讲解文字，辅助直观的算法流程图，编写缩进格式的实现程序，得到真实有效的运行结果。对于各种程序设计语言的共同概念，如数据类型、结构化程序设计的 3 种基本结构、数组及函数等进行深入讲解，使读者能够全面地理解程序设计语言，使读者能在此基础上自学其他程序设计语言。

全书共 12 章，第 1 章介绍 C 程序结构及其特点、Visual C++ 6.0 的安装及使用；第 2 章介绍数据类型、常量、变量、函数和表达式；第 3 章介绍传统流程图及 N-S 结构化流程图、赋值语句、数据输入输出函数调用语句及顺序结构程序设计；第 4 章介绍关系运算符、逻辑运算符和表达式、用 if 语句和 switch 语句实现选择结构程序设计；第 5 章介绍用 while 语句、do-while 语句和 for 语句实现循环结构程序设计及用 break 和 continue 语句提前结束循环；第 6 章介绍数组的定义和初始化、数组元素的使用、数值数组元素的常用操作、字符数组的使用；第 7 章介绍函数的定义、被调函数的声明、函数的调用、数组作函数参数、变量的作用域和存储类别；第 8 章介绍宏定义、文件包含、条件编译；第 9 章介绍结构体类型、共用体类型、枚举类型；第 10 章分别介绍指向变量的、指向数组的、指向函数的、指向指针的和指向结构体的指针变量及动态存储分配；第 11 章介绍文件的打开与关闭、文件的顺序读写和随机读写及文件检测函数；第 12 章介绍位运算符和位域。

除封面署名的作者外，在本书编写过程中，得到了所在学院的同事的热心帮助和支持，参加本书内容编写、程序调试、课件制作、习题收集、答案制作、内容审校等工作的老师还有李颖、于萍、逯洋、英昌盛、王海燕、叶丽娜、张桂杰、赵靖华、刘哲、李闯、张伟、赵鹏、王继魁、吕凯、刘伟、从飚、刘松、孙英慧、邹晓辉、王发斌、刘艳玲、李爽、罗琳、王晓雪等，在此向他们表示衷心的感谢。

需要本书电子课件、源文件及习题答案的读者，请到清华大学出版社网站下载。

由于时间仓促，书中难免存在不妥之处，请读者原谅，并提出宝贵意见。我们的电话是 010-62796045，信箱是 huchnehao@263.net。

编者

2012 年 12 月

目　录

第 1 章　引　言

人们与计算机进行交流是通过程序实现的，只有解决一定问题的程序才能指挥计算机自动地进行工作，而程序又是通过程序设计语言开发的。C 语言就是程序设计语言的一种，本章主要介绍 C 语言的程序结构及特点，重点介绍在 Visual C++ 6.0 环境中实现 C 程序功能的步骤。

1.1　程序设计语言

程序是指人们使用编程语言开发，为解决一定问题，能够被计算机执行的指令代码。计算机程序设计语言是编程人员应遵守的、计算机可以识别的程序代码规则，是人指挥计算机进行工作、与计算机进行交流的工具。

计算机程序设计语言是不断发展的。纵观其历史，可以将其分为低级语言和高级语言两大类。

1.1.1　低级语言

低级语言又称为面向机器的语言，因 CPU 的不同而不同，可移植性差。使用低级语言可以编出效率高的程序，但对程序设计人员的要求也很高。他们不仅要考虑解题思路，还要熟悉机器的内部结构，所以非专业人员很难掌握这类程序设计语言。

低级语言又分为机器语言和汇编语言。

1. 机器语言

机器语言是 CPU 可以直接识别的一组由 0 和 1 序列构成的指令代码。用机器语言编写程序，就是从所使用 CPU 的指令系统中挑选合适的指令，按照解决问题的算法组成一个指令序列。这种程序可以被机器直接理解并执行，速度很快，但因为它们不直观、难记、难写、不易查错、开发周期长，所以现在只有专业人员在编制对于执行速度有很高要求的程序时才采用。

2. 汇编语言

为了减轻编程者的劳动强度，人们使用一些帮助记忆的符号来代替机器语言中的 0、1 代码，使得编程效率和质量都有了很大的提高。由这些助记符组成的指令系统，称为符号语言，也称为汇编语言。汇编语言指令与机器语言指令基本上是一一对应的。因为这些助

记符不能被机器直接识别，所以用汇编语言编写的程序必须被汇编成机器语言才能被机器理解。汇编之前的程序称为源程序，汇编之后的程序称为目标程序。使用连接程序将目标程序连接成可执行程序。可执行程序能够脱离语言环境独立运行。

1.1.2 高级语言

高级语言提供大量的与人类语言相类似的控制结构，使程序设计者可以不关心机器的内部结构及工作原理，把主要的精力集中在解决问题的思路和方法上。这类摆脱了硬件束缚的程序设计语言的出现是计算机技术发展的里程碑，使得编程不再是少数专业人员的专利。由于高级语言不依赖具体的机器，所以用高级语言编写的程序可移植性好。

根据编程机制的不同，高级语言又分为面向过程的程序设计语言和面向对象的程序设计语言。

1. 面向过程的程序设计语言

面向过程的程序设计语言由一个入口和一个出口构成，程序每次执行都必须从这个入口开始，按照程序的结构执行到这个出口为止，属于过程驱动的编程机制，由过程控制程序运行的流向。编程人员要以过程为中心来考虑应用程序的结构，执行哪一部分代码和按何种顺序执行代码都由程序本身控制。它允许将程序分解为多个函数，这使得同一个程序可以由多人分工开发，大大提高了编程效率，使人们能够开发出规模越来越大、功能越来越强的应用软件和系统软件。常用的面向过程的语言有 C、Fortran、Pascal 等。

2. 面向对象的程序设计语言

面向对象的程序设计语言将整个现实世界或者其中的一部分看作是由不同种类的对象构成的，同一类型的对象既有相同点又有不同点。各种类型的对象之间通过发送消息进行联系，消息能够激发对象作出相应的反应，从而构成一个运动的整体，属于事件驱动的编程机制，由事件控制着程序运行的流向。编程人员要以对象为中心来设计模块，代码不是按预定的顺序执行，而是在响应不同的事件时执行不同的代码。当前使用较多的面向对象的程序设计语言有 Visual Basic、C++、C#、Java 等。

高级语言也不能被机器直接识别，也需要翻译才能运行。高级语言的运行方式有解释和编译两种。所谓解释，是指边解释边执行，不形成目标代码，执行速度不快，源程序保密性不强，如 Visual Basic 属于解释方式；所谓编译，是将源程序使用语言本身提供的编译程序编译为目标程序，再使用连接程序与库文件连接成可执行程序，可执行程序能够脱离语言环境独立运行。本课程所学的 C 语言程序设计属于编译方式。

1.2 程序结构及其特点

C 语言是国际上广泛流行的面向过程的结构化程序设计高级语言。C 语言是一种用途

广泛、功能强大、使用灵活的编程语言，既可用于编写应用软件，又能用于编写系统软件。

1.2.1 程序结构

计算机程序设计语言有不同的语法规则和程序结构，C 语言程序结构如图 1-1 所示。

1.2.2 程序结构的特点

通过分析图 1-1，可见 C 程序结构有以下几个特点。

(1) 一个 C 程序文件可以由一个或多个源文件(及零个或多个头文件)组成。

(2) 一个源文件或一个头文件可以依次包括 3 个部分：预处理指令、全局声明和函数定义。

(3) 一个源文件可由一个或多个函数组成，但有且仅有一个 main 函数，程序总是从 main 函数开始执行。

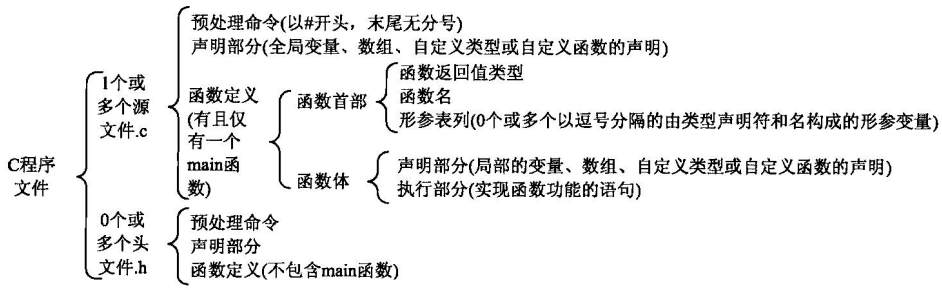

图 1-1 C 语言程序结构

(4) 一个头文件可由 0 个或多个函数组成，但不能有 main 函数。

(5) 一个函数的定义包括以下两部分。

① 函数首部：包括函数返回值类型、函数名、形参表列 3 个部分。其中，形参表列由形参类型及形参名构成。

② 函数体：包括声明部分和执行部分。其中，声明部分包括在本函数中所用到的变量或函数等的声明；执行部分由若干条语句组成，完成函数的功能。

(6) 一个声明或一个语句都必须以分号结尾，但预处理命令、函数首部等不加分号。

为了更好地说明 C 程序结构的特点，看以下两个程序，从中可以了解组成 C 程序的基本部分和书写格式。

【例 1-1】 在屏幕上输出以下信息“这是一个简单的 C 程序！”。

程序如下：

```
#include<stdio.h>                          // include 为文件包含预处理命令(以"#"开头)
void main()                                // main 是主函数的函数名
{
    printf("这是一个简单的 C 程序！\n");    //直接调用系统定义的库函数 printf
}
```

程序运行结果如图 1-2 所示。

程序分析：main 是主函数的函数名，每一个 C 程序都必须有且仅有一个 main 函数；在 main()之前的一行为预处理命令，这里的 include 称为文件包含命令，其意义是把尖括号<>或引号""内指定的文件包含到该程序中，成为该程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h，因此也称为头文件。C 语言的头文件中包括了各个标准库函数的函数原型，因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件；scanf 和 printf 是标准输入输出函数，其头文件为 stdio.h，在主函数前用 include 命令包含了 stdio.h 文件，直接调用即可。本例调用了输出函数 printf 把要输出的内容送到显示器显示。

【例 1-2】从键盘输入两个整数 x 和 y，求 x 和 y 的和，然后输出结果。

程序如下：

```
#include<stdio.h>                          //扩展名为.h 的文件称为头文件
void main()
{
    int x,y,s;                             //定义 3 个整型变量
    printf("input x:");                    //显示第一个提示信息
    scanf("%d",&x);                        //从键盘输入整数 x
    printf("input y:");                    //显示第二个提示信息
    scanf("%d",&y);                        //从键盘输入整数 y
    s=x+y;                                 //求 x 和 y 的和，并把它赋给变量 s
    printf("sum of %d and %d is %d\n",x,y,s);   //显示程序运算结果 s 的值
}
```

程序运行结果如图 1-3 所示。

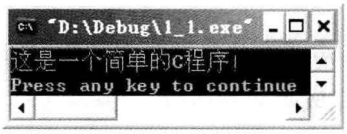

图 1-2　【例 1-1】的运行结果

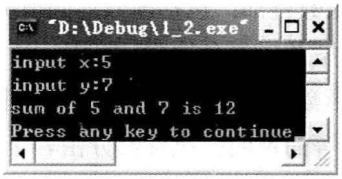

图 1-3　【例 1-2】的运行结果

程序分析：主函数体中又分为两部分，声明部分和执行部分。声明部分是 C 程序结构中很重要的组成部分。C 语言规定，程序中所有用到的变量都必须先声明，后使用，否则将会出错。【例 1-1】中未使用任何变量，因此无声明部分。本例中使用了 3 个变量 x，y 和 s，用来表示输入的自变量及求得的和。声明部分后的执行部分又称为执行语句部分，用以完成程序的功能。执行部分的第 1 行是输出语句，调用 printf 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第 2 行是输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 x 中。第 3 行是输出语句，调用 printf 函数在显示器上输出提示字符串，请操作人员输入自变量 y 的值。第 4 行是输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 y 中。第 5 行是求 x+y 的和，并把和送到变量 s 中。第 6 行是用 printf 函数输出变量 s 的值。

运行本程序时，首先在显示器屏幕上给出提示字符串 input x，这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数，如 5，按下回车键，然后在显示器屏幕上给出提示字符串 input y，这是由执行部分的第三行完成的。用户在提示下从键盘上键入某一数，如 7，按下回车键，接着在屏幕上显示计算结果 12。

1.2.3 程序书写规则

从书写清晰，便于阅读、理解和维护的角度出发，在书写程序时应遵循以下规则。

(1) 一行可以写多个声明或语句，但为了清晰，一个声明或一个语句最好占一行。每条声明或语句都有明确的含义，能完成一定的任务。

(2) 用{}括起来的部分，通常表示程序的某一层次结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

(3) 为了使程序便于阅读、易于调试，人们约定了锯齿形缩进的程序书写方式。将复合语句、函数体、循环体等语句用空格或 Tab 键向后缩进，使得程序错落有致，具有层次感。也就是说，低一层次的语句或声明比高一层次的语句或声明缩进若干格。

(4) 标识符和关键字之间至少加一个空格以示分隔。若已有明显的分隔符，也可不再加空格。

(5) C 语言声明或语句中使用的都是西方字符(称半角字符)，所以在输入源程序时，应该将中文输入法关闭，避免输入全角字母和符号。全角字母和符号只有在字符串常量中才可以使用，而且字母是区分大小写的。

(6) 程序中可以适当地加上注释，以增强程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程风格。

本书为了方便介绍语句、函数等的使用方法与语法格式，在命令格式中通常采用一些特殊的符号表示，如逗号加省略号、省略号等。这些符号不是命令的组成部分，在输入具体命令时，这些符号均不可作为语句中的成分输入计算机，它们只是命令的书面表示。具体含义如下。

,… 表示同类的项目重复多项。

… 表示省略了在当时叙述中不涉及的语句部分。

1.2.4 程序保留字

在 C 语言中使用的词汇分为 6 类：标识符、关键字、运算符、分隔符、注释符和常量，除标识符外，其他均为保留字，有特定的作用，不能挪作它用。

1. 关键字

关键字是由 C 语言规定的具有特定意义的字符串。C 语言的关键字分为以下几类。

(1) 类型声明符

用于定义(或声明)变量、数组、自定义函数或自定义数据类型，如 int、float、double 等。

(2) 语句定义符

用于表示一个语句的功能，如 if、for、while 等。

(3) 预处理命令字

用于表示一个预处理命令，如【例 1-1】和【例 1-2】中用到的 include。

2. 运算符

C 语言中含有丰富的运算符。运算符与常量、变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成，如算术运算符“+、-、*、/”等。

3. 分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型声明和函数参数表中，分隔各个变量；空格多用于语句各单词之间，作分隔符。在关键字、标识符之间必须要有一个以上的空格符作分隔，否则将会出现语法错误，例如，把“int a;”写成“inta;”，C 编译器会把 inta 当成一个标识符处理，其结果肯定出错。

4. 注释符

为了提高程序的可读性，通常在程序的适当位置加上必要的注释。C 语言的注释符有两种：一种是块注释，是以“/*”开头并以“*/”结尾的字符串；另一种是行注释，从“//”开始到行尾的字符串。注释可出现在程序中的任何位置，注释主要用来解释语句或函数的功能，用来向用户提示或解释程序的意义，以便他人或开发者日后能够读懂程序。程序编译时，不对注释作任何处理。在调试程序时可以对暂不使用的语句先用注释符括起来，使编译程序跳过处理，待调试结束后再去掉注释符。

5. 标识符

用来标识符号常量名、变量名、函数名、数组名、类型名、文件名等有效字符序列统称为标识符。除库函数的函数名由系统定义外，其余都由用户自己定义。

C 规定，标识符由字母(a~z，A~Z)、数字(0~9)、下划线(_)组成，并且第一个字符必须是字母或下划线，即标识符的命名规则是以字母或下划线开头的，后面跟着字母、数字或下划线的字符串。

在使用标识符时还必须注意以下几点。

(1) 标识符的长度受各种版本的 C 语言编译系统限制，同时也受具体机器的限制。

(2) 在标识符中，区分大小写。例如，b 和 B 是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义，但最好遵循见名知义的原则，便于阅读和理解。

1.3　Visual C++ 6.0 的安装及使用

按照 C 程序结构的要求，编制好了实现某一具体问题的程序，需要有相应的编程环境

来实现程序功能。目前，实现 C 编译系统有许多种，如 Visual C++ 6.0、Turbo C++ 3.0、Gcc 等，本书以 Visual C++ 6.0(简称为 VC++ 6.0)作为开发平台。

1.3.1 Visual C++ 6.0 的安装

Visual C++ 6.0 是 Visual Studio 套装软件中的一员，它可以和 Visual Studio 一起安装，也可以单独安装。运行安装光盘中的 setup.exe 文件，按照安装向导给出的提示，可以完成 Visual C++ 6.0 的安装。本书介绍的操作假定所使用的是单独安装的 Visual C++ 6.0。

1.3.2 Visual C++ 6.0 的启动

开机进入 Windows 后，可以用多种方法启动 Visual C++ 6.0。

方法 1：双击 Windows 桌面上的 Visual C++ 6.0 的快捷方式图标(如果桌面上有此快捷方式图标)，是最简单的启动方法。

方法 2：使用“开始”菜单中的程序命令。

如图 1-4 所示，单击 Windows 环境下的“开始”按钮，出现“开始”菜单，把鼠标指向“程序”菜单项，将出现“程序”子菜单；在“程序”子菜单中，把鼠标指向 Visual C++ 6.0 菜单项，出现 Visual C++ 6.0 子菜单，选择 Visual C++ 6.0 命令，即可进入 Visual C++ 6.0 编程环境，如图 1-5 所示。

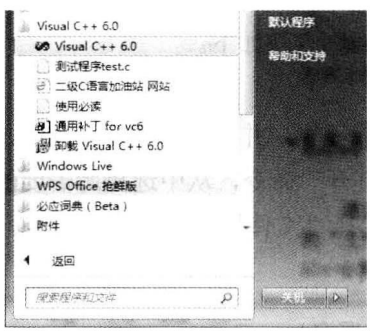

图 1-4 启动 Visual C++ 6.0

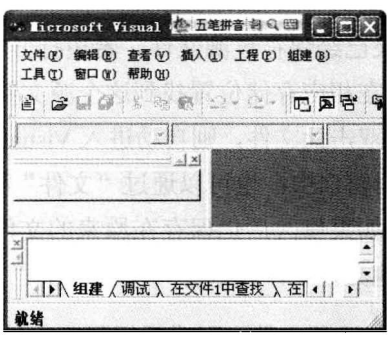

图 1-5 Visual C++ 6.0 主窗口

1.3.3 Visual C++ 6.0 上机过程

一个 C 程序的运行过程如下：编辑源文件(.c)，然后编译生成目标程序文件(.obj)，再将其链接生成可执行文件(.exe)，运行此文件得到程序结果。

1. 新建或打开一个源文件(.c 作为扩展名)

(1) 建立不带有工程的单个源文件

进入如图 1-5 所示的 Visual C++ 6.0 编程环境后，首先选择“文件”菜单下的“新建”命令，出现“新建”对话框，如图 1-6 所示，直接单击“新建”对话框的“文件”标签，打开“文件”选项卡，在列表框中选择 C++ Source File 选项，输入适当的位置及文件名，直接建立以.c 作为扩展名的源文件，在此文件的编辑窗口中输入和编辑源文件，并存盘，

编译时系统会提示建立工程工作空间，如图 1-7 所示。

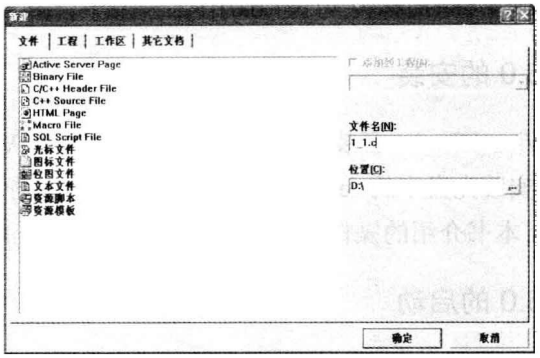

图 1-6　直接新建文件对话框

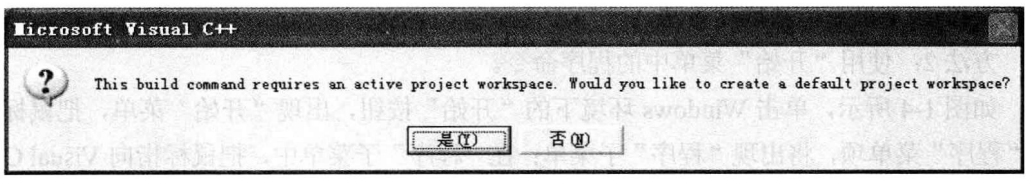

图 1-7　编译时提示新建工程工作空间的对话框

(2) 打开一个已有的源文件

如果已经保存过源文件，希望打开它进行编辑，则方法如下。

① 在相应存储位置找到这个源文件。

② 双击此文件，则自动进入 Visual C++ 6.0 集成环境，并打开该文件，此时，程序显示在编辑窗口中；也可以通过“文件”菜单下的“打开”命令，从中选择所需要的文件。

③ 如果修改后仍保存在原来的文件中，可以选择“文件”|“保存”命令，如果想存到其他文件中，则选择“文件”|“另存为”命令。

2. 编译源文件，形成目标程序文件(.obj 作为扩展名)

选择“组建”|“编译”命令，对程序进行编译，出现系统提示，单击“是”按钮，在调试信息窗口中会显示错误和警告等提示信息。若有错误或者警告，双击出错信息，即可在源文件中定位错误，此时，需要返回第 1 步修改程序源代码，再编译，直到没有错误，才可进行下一步操作。

3. 连接目标程序，形成可执行文件(.exe 作为扩展名)

若编译没有错误，则可使用“组建”|“组建”命令，在调试信息窗口中会显示错误和警告等提示信息。若有错误或者警告，需要返回第 1 步修改程序源代码，再调试，直到没有错误，则可进行下一步操作。

4. 执行可执行文件，得到程序运行结果

若组建没有错误，则直接进入结果窗口，显示最终结果。若结果不正确，则返回第 1

步修改程序源代码，再调试。

如果已完成一个程序的操作，不再对它进行其他处理，应当选择“文件”|“关闭工作空间”命令，结束对该程序的操作，为编辑下一个程序作准备。

5. 建立和运行包含多个文件的程序的方法

上面介绍的是最简单的情况，一个程序只包含一个源文件。如果一个程序包含多个源文件，则用户需要建立一个工程文件，在这个工程文件中包含多个文件(包括源文件和头文件)。工程文件是放在工程工作区中的，可由系统自动建立工程工作区。在编译时，系统会分别对工程文件中的每个文件进行编译，然后再将所得到的目标文件连接成为一个整体，再与系统的有关资源连接，生成一个可执行文件，最后执行这个文件，得到结果。建立和运行包含多个文件的程序有两种方法。

方法一：先建工程，再建文件。

(1) 选择“文件”|“新建”命令，出现如图 1-8 所示的“新建”对话框的“工程”选项卡，在列表框中显示了可以在 Visual C++ 6.0 中建立的工程类型，选择 Win32 Console Application 工程类型，然后选择合适的存储位置，并输入工程名称，选中“创建新的工作空间”单选按钮，单击“确定”按钮，创建一个空工程。

(2) 选择“文件”|“新建”命令，选择如图 1-9 所示的“新建”对话框的“文件”选项卡，在列表框中选择 C++ Source File 选项，输入适当的文件名，可在该工程中建立以.c 作为扩展名的源文件，在此文件的编辑窗口中输入和编辑源文件，并存盘；选择 C/C++ Header File 选项，输入适当的文件名，可在该工程中建立以.h 作为扩展名的头文件，在此文件的编辑窗口中输入和编辑头文件，并存盘。

(3) 编译和连接工程文件，生成可执行文件。

(4) 执行可执行文件。

方法二：先建文件，再建工程，将文件增加到工程中。

(1) 先利用图 1-6 所示对话框，分别编辑一个程序需要的各个文件(源文件文件以.c 作为扩展名，头文件以.h 作为扩展名)，并存放在指定的目录下。

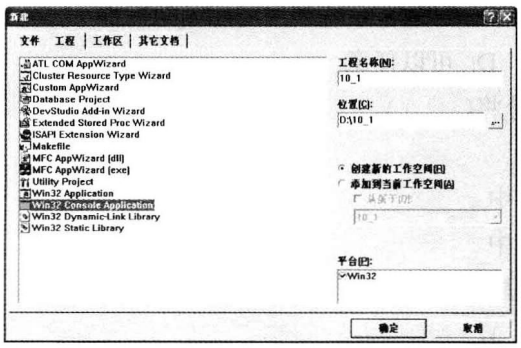

图 1-8 “新建工程”对话框

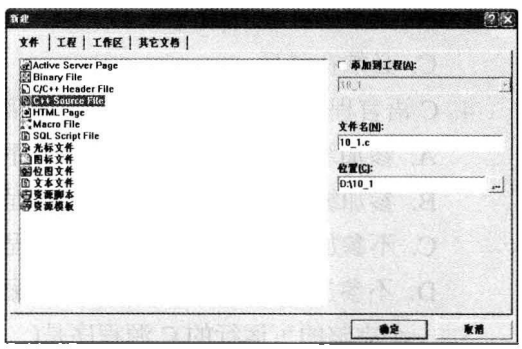

图 1-9 在工程中“新建文件”对话框

(2) 再利用图 1-8 所示对话框建立一个工程。

(3) 将各个文件加到工程中。方法是：选择“工程”|“增加到工程”子菜单下的“文