

C语言程序设计 一体化教程

李小遐◎主编



北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

C 语言程序设计

一体化教程

主 编 李小遐

副主编 刘慧梅 魏晓艳



 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书是C语言的入门教材,也是一部集理论讲授与上机实训为一体的理实一体化教材。全书共有11章,把C语言的学习分为三大部分。第1部分(第1、2章)介绍了C语言程序的基本框架、C语言程序的实现过程、数据类型及数据处理等程序设计的基础知识。第2部分(第3、4、5、6、7章)介绍了三种结构程序的设计、数组、函数等内容,培养程序设计的基本能力。第3部分(第8、9、10、11章)介绍了指针、结构体、文件,应用程序的开发过程等内容,培养设计应用程序的能力。

全书始终以程序实例驱动,将语法知识点同实际编程相结合,循序渐进,程序实例丰富有趣,便于教师讲授和学生学习。每章配有丰富的上机实训题目和课后习题,并且为其中的疑难题目设置了二维码,便于学生进行上机实践和课后复习巩固。

本书适合作为高等院校的教材,也适合作为其他人员学习C语言编程的入门书籍。

版权专有 侵权必究

图书在版编目(CIP)数据

C语言程序设计一体化教程/李小遐主编. —北京:北京理工大学出版社,2017.1
(2017.2重印)

ISBN 978-7-5682-0212-1

I. ①C… II. ①李… III. ①C语言-程序设计-高等职业教育-教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第003808号

出版发行/北京理工大学出版社有限责任公司

社 址/北京市海淀区中关村南大街5号

邮 编/100081

电 话/(010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址/<http://www.bitpress.com.cn>

经 销/全国各地新华书店

印 刷/三河市华骏印务包装有限公司

开 本/787毫米×1092毫米 1/16

印 张/16.5

字 数/388千字

版 次/2017年1月第1版 2017年2月第2次印刷

定 价/47.00元

责任编辑/钟 博

文案编辑/钟 博

责任校对/周瑞红

责任印制/李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前 言

本书是为高等院校编写的 C 语言学习教材，也是一本集理论讲授和上机实训为一体的理实一体化教材。针对高校学生的特点，编者特别强调在实践过程中学习 C 语言，领会程序编写的思路与方法，所以本书的编写思想是以程序案例驱动，将语法知识点同实际编程相结合，避免纠缠于语法细节，按照“提出问题→分析问题→用 C 语言程序解决问题→分析程序中的语法现象”的路径来讲解，这有利于培养学生分析问题与解决问题的能力，也使学生对语法的理解更为容易。

教学内容的安排是否合理，直接影响学生的学习效果。因此，本书特别注意前后内容的编排和衔接，以方便教师讲授和学生学习。本书各章按以下形式组织：

学习目标：为教师和学生规定明确的教学目标和学习目标；

学习内容：给出本章所有知识点；

授课内容：教师课堂讲授内容，为了弥补 C 语言语法的枯燥，本书配备大量的教学实例，而且特别注重这些实例的合理性和趣味性；

本章小结：总结本章的重要知识点，帮助学生整理复习；

实训：本章实训内容及指导，针对本章所介绍的语法知识，本书精心设计了上机实训内容指导，既方便教师布置上机实训作业，也便于学生上机前准备和上机后总结以及撰写实训报告。

习题：对本章内容的练习和巩固。

书中还设置了“提示”和“小测验”环节。容易出错的内容或特别需要说明的内容，以提示形式给出，比较醒目，容易记忆，小测验中提出的问题便于学生举一反三。

本书增加了二维码环节。针对实训和习题中的疑难内容设立了二维码，学生在独立思考后，可以通过手机扫码获取答案。

本书提供可直接使用的 PPT 电子教案，教师也可以根据需要进行修改后使用。

本书提供教学案例集，其中包括全书所有实例的源代码、各章实训源代码及习题源代码和答案。所有源代码均在 Windows 7 环境中，通过 Visual C++ 6.0（中文版）运行调试，而且所有程序实例的输出结果均为采用屏幕拷贝方式截取所得，充分体现了源代码的正确性。有需要的读者可以从北京理工大学出版社网站（www.bitpress.com）下载，或发邮件至邮箱 774016803@qq.com 索取。

本书由李小遐担任主编，负责全书的策划和统稿，由刘慧梅、魏晓艳担任副主编。其中李小遐编写第 1、2、3、5、7、11 章，刘慧梅编写第 6、8 章，魏晓艳编写第 4、9、10 章及附录。

本书在编写过程中参考了大量的文献资料，同时也得到北京理工大学出版社的大力协助，在此对参考文献的作者和北京理工大学出版社一并表示感谢。由于编者水平有限，书中不足之处恳请读者批评指正。

编 者

目 录

第1章 C语言程序的基本框架	1
1.1 初识C语言程序	1
1.1.1 C语言程序实例	1
1.1.2 C语言程序的组成	3
1.2 C语言程序的实现	5
1.2.1 C语言程序的实现过程	5
1.2.2 C语言程序的开发环境	6
1.3 本章小结	11
1.4 实训	12
习题1	13
第2章 基本数据类型及其运算	14
2.1 一个C语言程序实例	14
2.2 基本数据类型	15
2.3 常量和变量	16
2.3.1 常量	16
2.3.2 变量	18
2.4 运算符和表达式	19
2.4.1 算术运算符和算术表达式	20
2.4.2 赋值运算符和赋值表达式	22
2.4.3 自增自减运算	23
2.5 本章小结	24
2.6 实训	25
习题2	27
第3章 顺序结构及其应用	29
3.1 一个顺序结构程序实例	29
3.2 数据的输入/输出	30
3.2.1 格式输出函数	30
3.2.2 格式输入函数	33
3.2.3 字符输入/输出函数	34



3.3	顺序结构应用实例	35
3.4	本章小结	37
3.5	实训	38
	习题3	40
第4章	分支结构及其应用	42
4.1	一个分支结构程序实例	42
4.2	分支结构中条件的描述	43
4.2.1	关系运算符和关系表达式	43
4.2.2	逻辑运算符和逻辑表达式	44
4.3	实现分支结构的语句	45
4.3.1	if 语句	45
4.3.2	switch 语句	52
4.4	分支结构应用实例	55
4.5	本章小结	59
4.6	实训	59
	习题4	63
第5章	循环结构及其应用	68
5.1	一个循环结构程序实例	68
5.2	实现循环结构的语句	69
5.2.1	while 语句	69
5.2.2	do-while 语句	72
5.2.3	for 语句	74
5.3	循环嵌套	79
5.4	循环结构应用实例	82
5.5	本章小结	87
5.6	实训	87
	习题5	93
第6章	数组及其应用	97
6.1	一个 C 语言程序实例	97
6.2	一维数组	99
6.2.1	一维数组的定义和使用	99
6.2.2	一维数组应用举例	102
6.3	二维数组	104
6.3.1	二维数组的定义和使用	104
6.3.2	二维数组应用举例	108
6.4	字符串及其处理	110
6.4.1	字符串与字符数组	110



6.4.2	常用字符串处理函数	113
6.4.3	多个字符串的存储与处理	115
6.4.4	字符串应用举例	116
6.5	数组应用实例	118
6.6	本章小结	121
6.7	实训	122
	习题6	129
第7章	函数及其应用	134
7.1	一个C语言程序实例	134
7.2	函数的定义和调用	136
7.2.1	函数的定义	136
7.2.2	函数的调用	137
7.2.3	函数的声明	140
7.2.4	函数间的数据传递	141
7.3	变量的进一步讨论	142
7.4	函数应用实例	145
7.5	本章小结	150
7.6	实训	151
	习题7	155
第8章	指针及其应用	159
8.1	指针的基本用法	159
8.1.1	指针和指针变量的概念	159
8.1.2	指针变量的定义和使用	160
8.2	指针与数组	163
8.2.1	指针与一维数组	163
8.2.2	指针与二维数组	166
8.2.3	指针与字符串	169
8.3	指针与函数	170
8.3.1	变量地址作函数参数	170
8.3.2	数组名作函数参数	172
8.4	指针应用实例	174
8.5	本章小结	177
8.6	实训	178
	习题8	180
第9章	结构体及其应用	185
9.1	一个C语言程序实例	185
9.2	结构体类型的使用	187



9.2.1	结构体类型的定义	187
9.2.2	结构体变量的定义及引用	187
9.2.3	结构体数组	191
9.2.4	结构体指针	193
9.3	链表	195
9.3.1	链表的基本结构	195
9.3.2	链表的基本操作	196
9.3.3	链表综合应用	201
9.4	结构体应用实例	202
9.5	本章小结	206
9.6	实训	206
习题 9		210
第 10 章	文件及其应用	213
10.1	一个 C 语言程序实例	213
10.2	文件的概念	214
10.2.1	文件的概念及文件的类型	214
10.2.2	文件指针	214
10.3	文件的基本操作	215
10.3.1	文件的打开与关闭	215
10.3.2	文本文件的读写	216
10.3.3	二进制文件的读写	220
10.4	文件应用实例	224
10.5	本章小结	227
10.6	实训	228
习题 10		231
第 11 章	C 语言程序综合实训	233
11.1	应用程序设计方法	233
11.2	应用程序设计实例	234
11.3	C 语言课程实训	241
附录 A	常用字符与 ASCII 码对照表	246
附录 B	C 语言的库函数	248
附录 C	运算符的优先级和结合性	252
附录 D	C 语言的常用关键字	254
参考文献		256

第 1 章

C 语言程序的基本框架

学习目标

初识 C 语言程序，了解 C 语言程序的基本构成及构成元素，掌握 C 语言程序的开发过程及其开发环境。

学习内容

C 语言程序的组成、C 语言程序的实现过程及开发环境。

1.1 初识 C 语言程序

C 语言是一种面向解题过程的通用程序设计语言，其语言表达简洁、使用灵活，具有结构化的流程控制、丰富的数据结构和操作符、良好的程序可移植性和高效率的目标代码等特征，是最受编程初学者欢迎的计算机程序设计语言之一。

学习程序设计语言是使用计算机和研制计算机软件的必由之路，下面通过几个程序实例，进入 C 语言的精彩世界。

1.1.1 C 语言程序实例

【例 1-1】第一个 C 语言程序，在显示屏上显示信息 “This is my first C program!”。

【程序代码】

```
#include "stdio.h"
main()
{
    printf("This is my first C program!\n");
}
```

程序执行后的输出结果如图 1-1 所示。



图 1-1 【例 1-1】的输出结果

程序分析：

(1) 本程序非常简单，仅由 1 个 main() 主函数构成，在主函数中也只有 1 条语句，该语句通过标准输出函数 printf() 在显示屏上输出指定信息。



(2) 程序开头的“`#include "stdio.h"`”是预处理命令，其作用是包含输入/输出库文件，当程序中调用标准输入或输出函数时添加此行。

【例 1-2】计算并输出一个数的平方。

【程序代码】

```
#include "stdio.h"

main()                /* 主函数,程序从这里开始运行 */
{
    float a,b;         /* 函数体开始 */
    a=2.8;             /* 定义语句 */
    b=a*a;             /* 赋值语句 */
    printf("%f\n",b);  /* 赋值语句 */
    printf("%f\n",b);  /* 标准输出函数 */
}                      /* 函数体结束 */
```

程序执行后的结果如图 1-2 所示。



图 1-2 【例 1-2】的输出结果

程序分析：

(1) 程序中书写在“`/* */`”中的是注释，在程序中添加注释的目的是帮助程序阅读者阅读理解程序。必要的注释可以增加程序的可读性，但是注释对程序的执行没有任何影响，编译时将被过滤掉，因此注释可以添加在程序的任何位置。

(2) 本程序仍然由 1 个主函数构成，相对【例 1-1】稍复杂的是，主函数中含有 4 个不同的语句，共同完成程序要求的计算功能。

【例 1-3】从键盘任意输入两个数，输出其中的大数。

【程序代码】

```
#include "stdio.h"

int max(int x, int y)    /* 定义 max() 函数,求两个数中的大数 */
{
    int z;
    if(x>y) z=x;else z=y;
    return(z);          /* 返回函数值 */
}

main()                  /* 主函数,程序从这里开始执行 */
{
    int a,b,c;
    printf("请输入两个整数:");
    scanf("%d,%d",&a,&b); /* 标准输入函数 */
```



```
c = max(a,b); /* 调用 max() 函数 */  
printf("%d , %d 中的大数为: %d\n\n", a, b, c); /* 标准输出函数 */  
}
```

程序执行时, 如果从键盘提供两个整数 12 和 21, 得到的输出结果图 1-3 所示。



```
请输入两个整数: 12, 21  
12 , 21 中的大数为: 21
```

图 1-3 【例 1-3】的输出结果 (1)

程序执行时, 如果从键盘提供的两个整数是 31 和 13, 那么得到的输出结果如图 1-4 所示。



```
请输入两个整数: 31, 13  
31 , 13 中的大数为: 31
```

图 1-4 【例 1-3】的输出结果 (2)

程序分析:

(1) 本程序由两个函数模块组成: 主函数 `main()` 和子函数 `max()`。子函数 `max()` 称为自定义函数, 其功能是找出两个数中的大数。

(2) 标准库函数由 C 系统定义, 用户在程序中直接调用即可, 比如程序中的输入函数 `scanf()` 和输出函数 `printf()`。与标准库函数不同, 自定义函数由用户定义, 一旦定义好, 就可以像标准库函数一样使用。有关自定义函数的内容, 本书将在第 7 章介绍。

(3) 程序从 `main()` 函数开始执行, 执行到语句 “`c = max(a, b);`” 时转到 `max()` 函数, 遇到 `return` 语句返回主函数继续执行, 直到程序结束。

1.1.2 C 语言程序的组成

以上实例及实例分析已经反映出了 C 语言程序的基本框架及其组成要素。

1. C 语言程序是由函数构成的

C 语言程序通常由包括 `main()` 在内的一个或多个函数组成, 函数是构成 C 语言程序的基本单位。其中, 主函数必须有且只能有一个, 被调用的其他函数可以是系统提供的库函数, 也可以是用户自定义的函数。C 语言程序的全部工作都是由函数来完成的, 因此 C 语言被称为函数式语言。

2. 函数的构成

C 语言的函数由函数首部和函数体两大部分组成。以主函数为例:

```
main() ← 函数首部, 指定函数名、函数参数等信息  
{      ← 函数体从这里开始  
..... ← 函数体内的语句  
}      ← 函数体到这里结束
```



3. 语句

语句是构成函数的基本单位，函数功能的实现由若干条语句序列完成。程序中的语句有说明语句和执行语句，说明语句完成数据的描述，执行语句完成指定的操作功能。每个语句必须以“;”结束，这是 C 语言的一个特色。

提示：一个 C 语言程序由若干条 C 语句组成，每个 C 语句完成一个特定的操作。

4. 标识符

标识符是程序中函数、变量、语句及数据类型等对象的名称。C 语言的标识符可分为两类：

(1) 关键字。关键字是 C 语言系统规定的、具有特定含义和专门用途的一些字符序列，如前面程序中出现的 `main`、`int`、`printf`、`scanf`、`include`、`return` 等。

在程序中使用关键字时，不能用错，也不能将之挪作他用。

(2) 用户标识符。用户标识符是设计程序时用户自己定义的名字，这类标识符命名时要遵守以下规则：由字母（26 个大、小写字母）、数字（0~9）和下划线（`_`）组成，不能以数字开头，区分大、小写。为了便于使用，命名应简洁、实用。

小测验

以下标识符中，哪些是合法标识符？哪些是不合法标识符？

<code>int</code>	<code>define</code>	<code>double</code>	<code>switch</code>	<code>while</code>
<code>m + y</code>	<code>a#</code>	<code>b - 4</code>	<code>π</code>	<code>2x</code>
<code>_1</code>	<code>Abc</code>	<code>_max</code>	<code>b_1</code>	<code>a123</code>
<code>name</code>	<code>a</code>	<code>a1</code>	<code>sum</code>	<code>aver</code>

提示：

- (1) 标识符中不能含有除字母、数字和下划线外的其他字符。
- (2) 用户标识符不能与保留字重名。
- (3) 标识符命名以“顾名思义”为佳。

5. C 语言程序的书写规范

C 语言程序的书写没有格式要求，因此可以自由书写。但是为了程序的易读性，长期以来人们还是形成了一定的规范。

(1) C 语言没有行的概念，但是通常一行写一条语句，当然一行可以写多条语句，一条语句也可以写在多行上。

(2) 整个程序采用缩进格式书写，表示同一层次的语句行对齐，缩进同样多的字符位置，比如选择体和循环体中的语句要缩进对齐。

(3) 程序代码习惯用小写字母，只是在特定的时候才使用大写字母。

(4) 在程序中恰当地使用空行，分隔程序中的语句块，以增加程序的可读性。

(5) 在程序中加注释有助于阅读、理解程序。



良好的程序设计风格有助于编写出既可靠又易维护的程序，可保证程序的结构清晰、合理。在学习程序设计初期，应注意培养良好的书写风格，养成一种良好的代码书写习惯，减少编程中低级错误的出现，从而提高编程及调试程序的效率。

比如：以下两个程序都能编译成功，但你更愿意阅读哪个程序？

```
#include "stdio.h"
main()
{
    printf("有趣的程序");
}
```

```
#include "stdio.h"
main()
{
    printf("有趣的程序");}
```

提示：C语言程序中花括号“{}”使用得比较多，书写程序时要注意“{”和“}”应成对使用。

1.2 C语言程序的实现

1.2.1 C语言程序的实现过程

按照C语言语法规则编写的C语言程序称为源程序。设计好源程序后，要将其输入计算机并得到最终结果，必须经过编辑、编译、连接和运行这几个主要环节，其实现过程如图1-5所示。

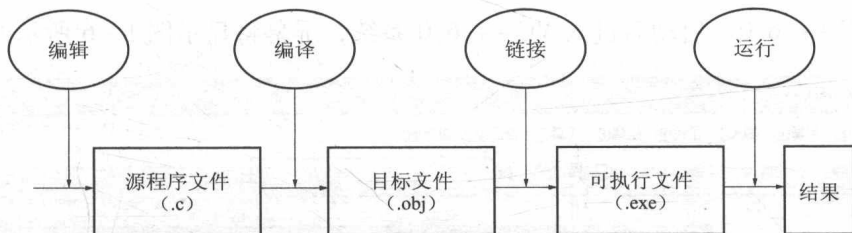


图1-5 C语言程序的实现过程

1. 编辑源程序

对设计好的源程序，要利用程序编辑器将其输入到计算机中，输入的程序一般以文本文件的形式存放在磁盘上，文件的扩展名为“.c”。所用的编辑器可以是任何一种文本编辑软件，比如Turbo C和VC++6.0这样的专用编辑系统，Windows系统提供的写字板或文字处理软件等也可以用来编辑源程序。

2. 编译源程序

源程序是无法直接被计算机执行的，因为计算机只能执行二进制的机器指令，这就需要把源程序先翻译成机器指令，然后计算机才能执行翻译好的程序，这个过程是由C语言的编译系统完成的。源程序编译之后生成的机器指令程序叫目标程序，其扩展名为“.obj”。



3. 连接程序

在源程序中, 输入、输出等标准函数不是用户自己编写的, 而是直接调用系统函数库中的库函数。因此, 必须把目标程序与库函数进行连接, 才能生成扩展名为 “. exe” 的可执行文件。

4. 运行程序

执行 “. exe” 文件, 得到最终结果。

在编译、连接和运行程序的过程中, 有可能出现错误, 此时可根据系统给出的错误提示对源程序进行修改, 并重复以上环节, 直到得出正确的结果为止。

1.2.2 C 语言程序的开发环境

C 语言是一种编译型程序设计语言, 因此, 一个 C 语言程序需要经过编辑、编译、连接和运行四个步骤才能得到运行结果。由于 C 语言已经是一种比较成熟的程序设计语言, C 语言的标准已被大多数 C \ C ++ 的开发环境所兼容, 所以 C 语言程序可以在多种编程环境中被开发和运行。目前使用的大多数 C 语言编译系统都是集成开发环境 (IDE), 即把程序的编辑、编译、连接和调试运行等操作集成在一起, 功能丰富、操作方便, 比较适合初学者使用。Microsoft Visual C ++ 6.0 (简称 VC ++ 6.0) 就是一款开发 C 语言程序的集成开发环境, 运行于 Windows 操作系统, 是目前比较常用的一款开发工具。下面以 VC ++ 6.0 (中文版) 为编辑平台, 介绍 C 语言程序的实现过程。

1. 启动 VC ++

安装 VC ++ 6.0, 启动后进入 VC ++ 6.0 系统, 屏幕将显示图 1-6 所示的窗口。

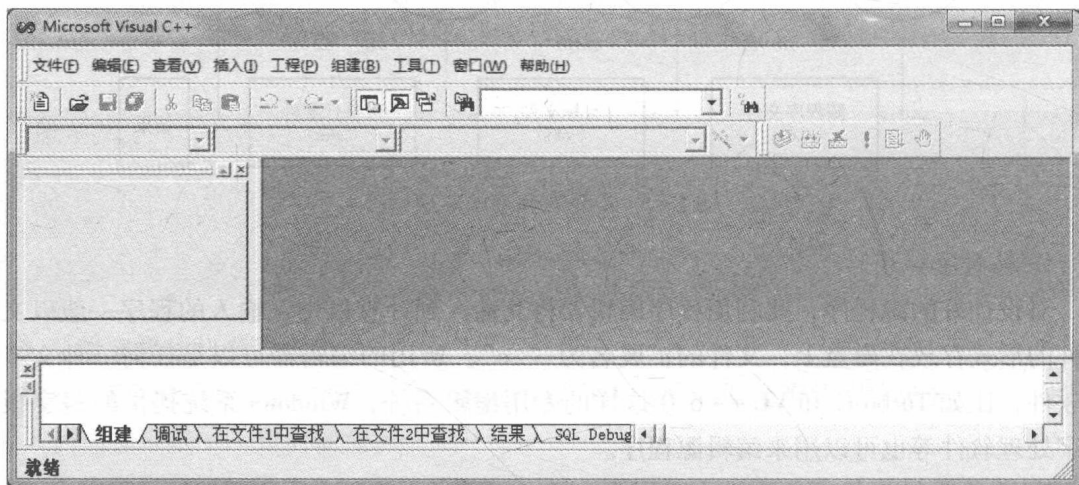


图 1-6 VC ++ 6.0 主窗口

2. 新建 C 语言程序文件

在图 1-6 中选择 “文件” 菜单的 “新建” 菜单项, 会弹出 “新建” 窗口, 如图 1-7 所示。单击 “文件” 标签, 选中 “C ++ Source File”, 同时在右边文件输入框中输入文件名, 比如 “a01. c”, 在位置框中选择或输入文件保存的位置, 然后按 “确定” 按钮。

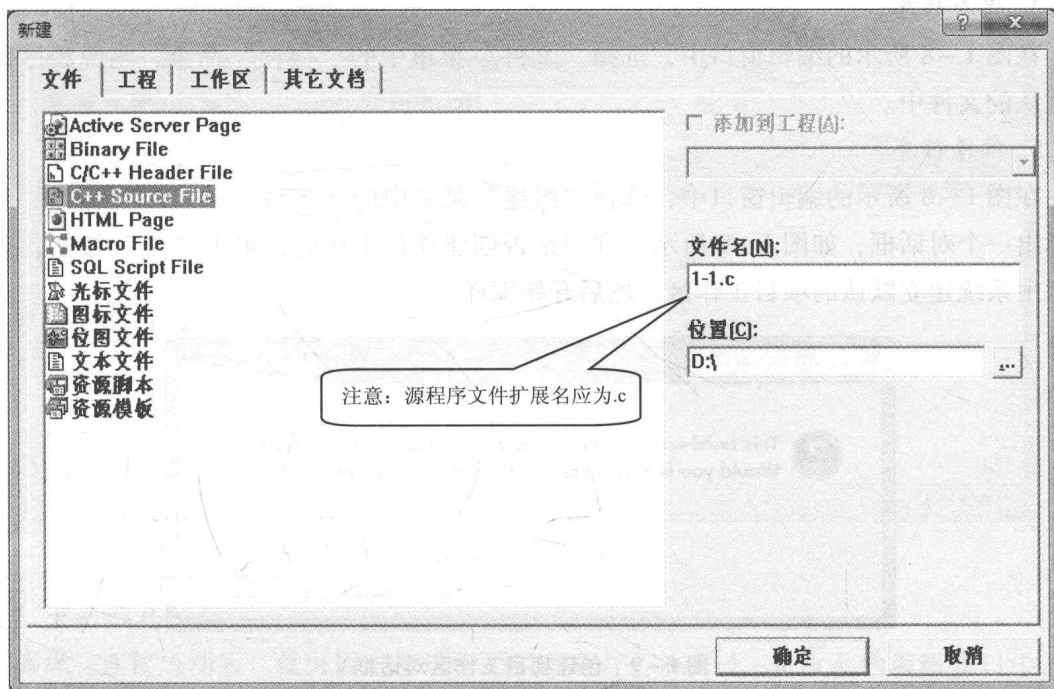


图 1-7 “新建”窗口

提示:VC++集成环境约定,当源程序文件的扩展名为“.c”时,则为C语言程序,而当源程序文件的扩展名为“.cpp”时,则为C++程序。

3. 编辑源程序

在图 1-8 所示的编辑窗口输入程序代码。由于其完全是 Windows 界面,输入及修改可借助鼠标和菜单进行,十分方便。



图 1-8 源程序编辑界面



4. 保存程序

在图 1-8 所示的编辑窗口中, 选择“文件”菜单中的“保存”命令, 将源程序保存到默认的文件中。

5. 编译程序

在图 1-8 所示的编辑窗口中, 选择“组建”菜单中的“编译 [a01.c]”命令, 此时将弹出一个对话框, 如图 1-9 所示, 询问是否创建项目工作区, 单击“是”按钮, 表示同意由系统建立默认的项目工作区, 然后开始编译。

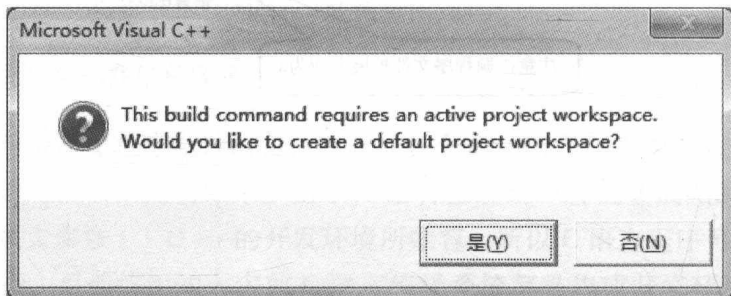


图 1-9 创建项目工作区对话框

在编译时, 编译系统将检查源程序中是否有语法错误, 然后在主窗口下部的调试信息窗口输出编译信息。如果发现错误, 编译信息就会给出错误的位置和性质; 如果代码编译没有错误, 则在信息调试窗口显示图 1-10 所示的信息, 表示编译成功, 生成目标文件, 比如“a01.obj”。

在编译成功后, 程序还不能直接运行, 还需要把目标程序 (.obj) 与系统提供的资源 (如函数库、头文件等) 连接成为可执行程序 (.exe)。

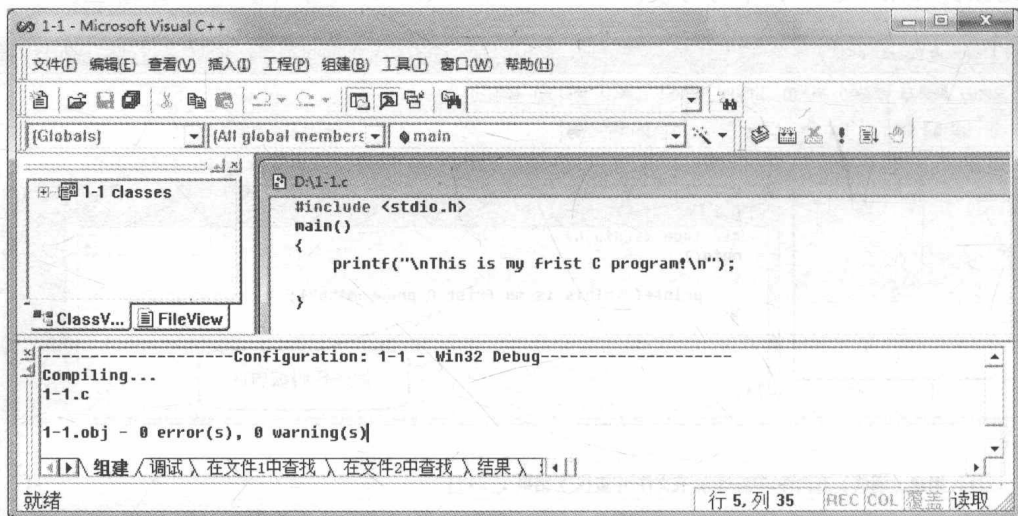


图 1-10 编译成功窗口

6. 连接程序

在图 1-8 所示的编辑窗口中, 选择“组建”菜单中的“组建 [a01.exe]”命令, 连接成功后即可生成一个可执行程序, 如“a01.exe”, 其界面如图 1-11 所示。

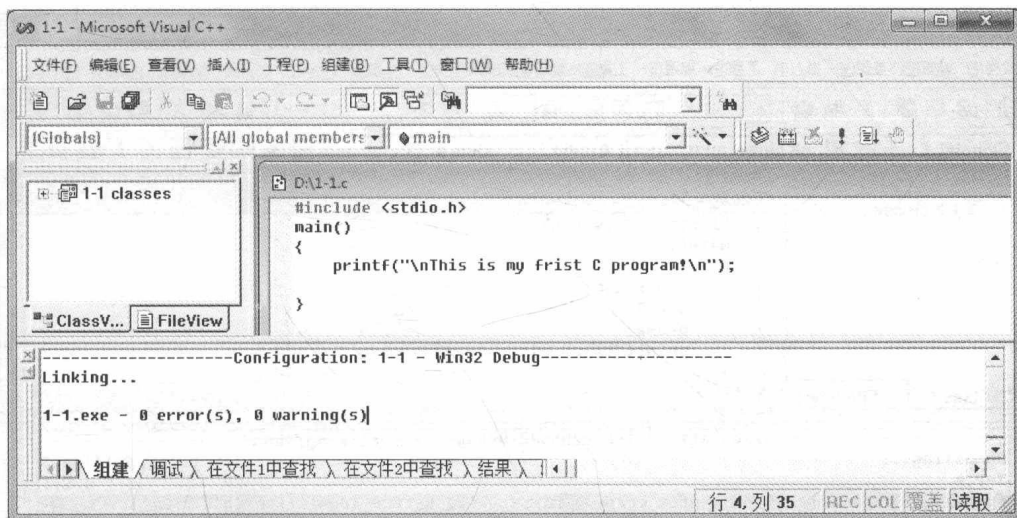


图 1-11 连接成功窗口

7. 运行程序

编译、连接成功后，就可以直接运行可执行程序。在图 1-8 所示的编辑窗口中，选择“组建”菜单中的“! 执行 [a01.exe]”命令，即可运行程序，此时窗口自动切换到结果窗口，显示程序的运行结果，如图 1-12 所示。

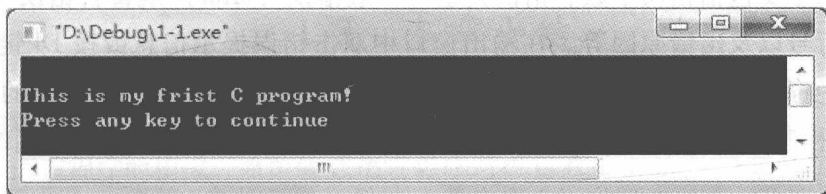


图 1-12 程序运行结果窗口

对于编译、连接和运行操作，VC++6.0 还提供了一组快捷工具按钮，如图 1-13 所示，单击该工具按钮可快速进行操作。



图 1-13 快捷工具按钮

8. 调试程序

调试程序是程序设计中一个很重要的环节，一个程序很难保证一次就能运行通过，一般都要经过多次调试。程序中的错误一般分为源程序语法错误和程序设计上的逻辑错误，编译时只能找出语法错误，而逻辑错误需要靠程序员手工查找。

如果程序中存在语法错误，那么编译时会在输出窗口中给出错误提示，如图 1-14 所示。