



全国高等职业教育规划教材

C语言程序设计教程

主 编 葛素娟 胡建宏

副主编 罗大伟 雷 晖



电子教案、源代码下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

C 语言程序设计教程

主 编	葛素娟	胡建宏	
副主编	罗大伟	雷 晖	
参 编	叶冬鲜	罗红艳	张 鹏
主 审	朱祥贤		



机械工业出版社

本书是根据高职高专的培养目标,结合高职高专教学改革和课程改革的要求,本着“理实一体、教学做结合”的原则编写的。

本书主要介绍 C 语言数据类型、运算符及表达式,顺序结构、选择结构、循环结构程序设计,数组,函数,预处理命令,指针,结构体、共用体及枚举,位运算,文件以及应用项目、习题、实验等内容。

本书可以满足单片机、Java、嵌入式系统等后续课程对 C 语言的要求,结构清晰,语言简练,图文并茂,具有很强的实用性和可操作性,可作为电子、通信、计算机、机电类各专业 C 语言程序设计的教材,也可作为相关专业学生的自学参考书和培训教材。

本书配套授课电子课件、源程序、习题及实验解答,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 1239258369, 电话: 010-88379739)。

图书在版编目(CIP)数据

C 语言程序设计教程/葛素娟,胡建宏主编.—北京:机械工业出版社,2013.1

全国高等职业教育规划教材

ISBN 978-7-111-41090-4

I. ① C… II. ① 葛… ② 胡… III. ① C 语言-程序设计-高等职业教育-教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2013)第 007452 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:鹿征 刘闻雨

责任印制:乔宇

保定市中国画美凯印刷有限公司印刷

2013 年 2 月第 1 版·第 1 次印刷

184mm×260mm·19.25 印张·476 千字

0001~3000 册

标准书号:ISBN 978-7-111-41090-4

定价:39.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

社服务中心:(010)88361066

销售一部:(010)68326294

销售二部:(010)88379649

读者购书热线:(010)88379203

网络服务

教材网:<http://www.cmpedu.com>

机工官网:<http://www.cmpbook.com>

机工官博:<http://weibo.com/cmp1952>

封面无防伪标均为盗版

目 录

出版说明

前言

第1章 C语言概述 1

1.1 C语言的发展概况 1

1.2 C语言的特点 1

1.3 C程序的工作环境

Visual C++ 6.0 2

1.3.1 启动 Visual C++ 6.0 2

1.3.2 C程序的编译、连接和执行 6

1.4 简单的C程序 7

1.5 打地鼠游戏程序之一 11

1.6 习题一 13

第2章 数据类型、运算符及表达式 15

2.1 C语言的数据类型 15

2.2 常量和变量 15

2.2.1 常量 15

2.2.2 变量 19

2.3 变量的初始化 26

2.4 各类数值型数据间的混合运算 26

2.4.1 自动转换 26

2.4.2 强制类型转换 27

2.5 算术运算符和算术表达式 27

2.5.1 C语言运算符简介 27

2.5.2 算术运算符和算术表达式 28

2.6 赋值运算符和赋值表达式 31

2.6.1 简单赋值运算符 31

2.6.2 复合赋值运算符 31

2.7 逗号运算符和逗号表达式 33

2.8 求字节运算符 34

2.9 习题二 35

第3章 顺序结构程序设计 38

3.1 C语句概述 38

3.2 流程图 39

3.3 顺序结构程序设计 40

3.4 字符数据的输入输出 41

3.4.1 putchar()函数 41

3.4.2 getchar()函数 41

3.5 格式输入输出 42

3.5.1 printf()函数 42

3.5.2 scanf()函数 44

3.5.3 使用printf()函数和scanf()函数时的
注意事项 46

3.6 程序举例 46

3.7 习题三 47

第4章 选择结构程序设计 50

4.1 关系运算符与关系表达式 50

4.1.1 关系运算符 50

4.1.2 关系表达式 50

4.1.3 关系表达式的值 50

4.2 逻辑运算符与逻辑表达式 51

4.2.1 逻辑运算符 51

4.2.2 逻辑表达式 51

4.2.3 逻辑表达式的值 52

4.3 if语句 53

4.3.1 单分支if语句 53

4.3.2 双分支if...else...语句 55

4.3.3 多分支if...else if语句 58

4.3.4 if语句的嵌套 59

4.3.5 条件运算符与条件表达式 60

4.4 switch语句 61

4.5 程序举例 64

4.6 习题四 67

第5章 循环结构程序设计 71

5.1 循环语句 71

5.1.1 for 语句	72	7.7 局部变量和全局变量	142
5.1.2 while 语句	76	7.7.1 局部变量	142
5.1.3 do while 语句	77	7.7.2 全局变量	143
5.1.4 while 语句和 do while 语句的 比较	77	7.8 变量的存储类别	145
5.2 循环的嵌套	78	7.8.1 动态存储方式与静态存储 方式	145
5.3 两种转移控制语句	80	7.8.2 auto 变量	145
5.3.1 break 语句	80	7.8.3 用 static 声明局部变量	146
5.3.2 continue 语句	81	7.8.4 register 变量	147
5.4 程序举例	82	7.8.5 用 extern 声明外部变量	148
5.5 习题五	85	7.8.6 用 static 声明外部变量	150
第6章 数组	90	7.9 内部函数和外部函数	151
6.1 一维数组	90	7.9.1 内部函数	151
6.1.1 一维数组的定义	91	7.9.2 外部函数	151
6.1.2 一维数组元素的引用	92	7.10 程序举例	153
6.1.3 一维数组的初始化	93	7.11 打地鼠游戏程序之三	154
6.2 二维数组	100	7.12 习题七	157
6.2.1 二维数组的定义	100	第8章 预处理命令	161
6.2.2 二维数组元素的引用	101	8.1 宏定义	161
6.2.3 二维数组的初始化	103	8.1.1 不带参数的宏定义	161
6.3 字符数组	106	8.1.2 带参数的宏定义	163
6.3.1 字符数组的定义	107	8.2 “文件包含”处理	166
6.3.2 字符数组元素的引用	107	8.3 条件编译	167
6.3.3 字符数组的初始化	108	8.4 习题八	169
6.3.4 字符串处理函数	111	第9章 指针	171
6.4 程序举例	119	9.1 指针的概念	171
6.5 打地鼠游戏程序之二	122	9.2 变量的指针和指向变量的 指针变量	173
6.6 习题六	125	9.2.1 指针变量的定义	173
第7章 函数	130	9.2.2 指针变量作为函数参数	175
7.1 函数简述	130	9.3 数组的指针和指向数组的 指针变量	176
7.2 函数定义的一般形式	132	9.3.1 数组的指针	176
7.2.1 无参函数定义的一般形式	132	9.3.2 数组元素的指针	176
7.2.2 有参函数定义的一般形式	133	9.3.3 通过指针引用数组元素	177
7.3 函数参数和函数返回值	133	9.3.4 用数组名作函数参数	179
7.4 函数的嵌套调用	136	9.4 字符串的指针和指向字符串的	
7.5 函数的递归调用	138		
7.6 数组名作函数参数	141		

指针变量	183	10.9.2 共用体类型的定义	230
9.4.1 字符串的表示形式	183	10.9.3 共用体变量的定义	230
9.4.2 用字符数组和字符串指针作 函数参数	184	10.9.4 共用体变量的引用	232
9.5 函数的指针和指向函数的 指针变量	186	10.10 枚举类型	233
9.5.1 函数的指针	186	10.10.1 枚举类型的定义	233
9.5.2 指向函数的指针变量	186	10.10.2 枚举变量的定义	234
9.6 返回指针值的函数	187	10.10.3 枚举变量的引用	234
9.7 指针数组和指向指针的指针	188	10.11 用 typedef 定义类型	238
9.7.1 指针数组	188	10.12 电动车电池管理程序之一	238
9.7.2 指向指针的指针	190	10.13 习题十	244
9.7.3 指针数组作 main 函数的形参	190	第 11 章 位运算	249
9.8 指向一维数组的指针变量	192	11.1 位运算的概念	249
9.9 习题九	193	11.2 位运算符	249
第 10 章 结构体、共用体及枚举	197	11.2.1 位逻辑运算符	249
10.1 结构体的概念	197	11.2.2 移位运算符	252
10.2 结构体类型的定义	198	11.3 位运算的常用用法	254
10.3 结构体变量的定义	199	11.4 位运算复合赋值运算符	254
10.4 结构体变量的引用	201	11.5 位段	254
10.5 结构体变量的初始化	203	11.6 习题十一	256
10.6 结构体数组	204	第 12 章 文件	258
10.6.1 结构体数组的定义	205	12.1 文件概述	258
10.6.2 结构体数组的初始化	206	12.1.1 文件的概念与文件的存储	258
10.7 指向结构体类型数据的指针	208	12.1.2 文件的存取方式	259
10.7.1 指向结构体变量的指针	208	12.1.3 C 语言的设备文件	259
10.7.2 指向结构体数组的指针	210	12.2 文件指针	259
10.7.3 指向结构体的指针作函数 参数	211	12.2.1 FILE 类型	259
10.8 用指针处理链表	214	12.2.2 文件指针	260
10.8.1 建立简单链表	214	12.3 文件的打开与关闭	260
10.8.2 处理动态链表所需的函数	217	12.3.1 文件的打开	260
10.8.3 建立动态链表	218	12.3.2 文件的关闭	261
10.8.4 对链表的删除	221	12.4 文件的读写	261
10.8.5 对链表的插入	224	12.4.1 单个字符的读写	261
10.9 共用体	229	12.4.2 字符串的读写	264
10.9.1 共用体的概念	229	12.4.3 数据块的读写	266
		12.4.4 格式化读写	270
		12.5 文件的定位	271
		12.5.1 rewind 函数	272

12.5.2	fseek 函数	273	实验六	一维数组	287
12.5.3	ftell 函数	273	实验七	二维数组	287
12.6	出错的检测	274	实验八	字符数组	288
12.6.1	ferror 函数	274	实验九	函数	288
12.6.2	clearerr 函数	275	实验十	指针	288
12.7	电动车电池管理程序之二	276	实验十一	结构体	288
12.8	习题十二	282	实验十二	文件	289
实验部分		285	附录		290
实验一	熟悉 C 语言	285	附录 I	常用字符与 ASCII 代码 对照表	290
实验二	数据类型、运算符及 表达式	285	附录 II	运算符的优先级和结合性 ...	292
实验三	顺序结构程序设计	285	附录 III	常用 C 库函数	293
实验四	选择结构程序设计	286	附录 IV	C 语言中的关键字	297
实验五	循环结构程序设计	286	参考文献		298

第1章 C语言概述

本章要点

- 了解 C 语言的发展概况、特点等相关内容。
- 熟悉 C 程序的工作环境 Visual C++ 6.0。
- 学会使用 `#include <stdio.h>`，掌握 `main()`、`printf()`、`scanf()` 的基本用法，并且能够编写出简单的 C 程序。

1.1 C语言的发展概况

C 语言是 20 世纪 70 年代初由美国贝尔实验室在 B 语言的基础上发展起来的，它保持了 B 语言精练、接近硬件的特点，又改进了 B 语言过于简单的缺点。早期的 C 语言主要是在贝尔实验室内部使用。1978 年以 C 语言编译程序为基础，B·W·Kernighan 和 D·M·Ritchie 合著了著名的《The C Programming Language》一书。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础。但是，在这本书中并没有定义一个完整的标准 C 语言，后来由美国国家标准协会（ANSI）在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

早期的 C 语言主要是应用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了 20 世纪 80 年代，C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

在 C 语言的基础上，贝尔实验室在 1983 年推出了 C++。C++ 进一步扩展和完善了 C 语言，成为一种面向对象的程序设计语言。

1.2 C语言的特点

C 语言是一种通用、灵活、结构化和使用普遍的计算机高级语言，既可作为系统软件的描述语言，也可用来开发应用程序，特别适合进行系统程序设计和对硬件进行操作的场合。

C 语言的主要特点如下。

(1) C 语言简洁紧凑，使用方便

标准 C 语言（ANSI C）只有 32 个关键字，9 种控制语句。书写形式自由，一行可以书写多条语句，一个语句也可以写在不同行上。

(2) C 语言介于汇编语言与高级语言之间

C 语言既像汇编语言那样允许直接访问物理地址，能进行位运算，能实现汇编语言的大部分功能，直接对硬件访问，也有高级语言面向用户、易记忆、易学习及易于书写的特点。

(3) C 语言是一种结构化语言

C 语言的主要成分是函数。函数是 C 语言程序的基本结构模块，程序可以由不同功能的

函数有机组装而成，从而可以达到结构化程序设计中模块的要求。另外，C 语言提供了 3 种基本结构（顺序、选择、循环），使程序流程具有良好的结构性。

(4) C 语言有丰富的数据类型

C 语言具有现代化语言的各种数据类型。用户能自己扩充数据类型，实现各种复杂的数据结构，完成用于具体问题的数据描述。

(5) C 语言具有较高的移植性，目标代码质量高、运行效率高

在 C 语言中，没有专门与硬件有关的输入/输出语句，程序的输入/输出通过调用库函数实现，使 C 语言本身不依赖于硬件系统，用 C 语言写的程序基本上不做修改就能用于各种型号的计算机和各种操作系统，这大大提高了程序的可移植性。

用 C 语言编写的程序，其生成的目标代码质量高、运行效率高，一般只比汇编程序生成的目标代码效率低 10% ~ 20%。

C 语言的以上特点，读者现在也许还不能深刻理解，待学完 C 语言以后再回顾一下，就会有比较深的体会。

1.3 C 程序的工作环境 Visual C ++ 6.0

Visual C ++ 6.0 是一款功能强大的，面向对象的程序设计语言，是 C ++ 的版本，而 C ++ 语言是在 C 语言的基础上扩展而成的，所以 C 程序也能在该环境下运行，但仅是 Visual C ++ 6.0 强大功能中的很少部分的应用。或者说，Visual C ++ 6.0 包含 C，C 是 Visual C ++ 6.0 的子集。

1.3.1 启动 Visual C ++ 6.0

启动 Visual C ++ 6.0 步骤如下。

1) 在 Windows 环境下选择“开始”→“程序”→“Microsoft Visual C ++ 6.0”→“Microsoft Visual C ++ 6.0”，如图 1-1 所示。

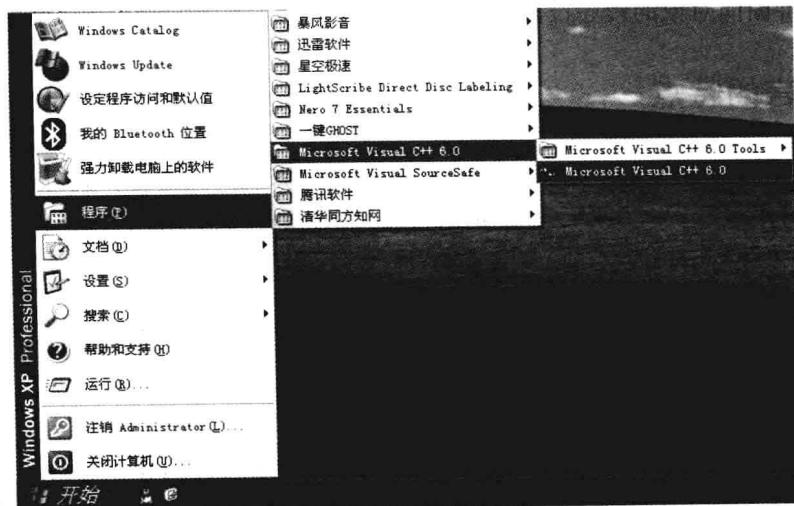


图 1-1 启动 Visual C ++ 6.0

2) 出现 Visual C++ 6.0 集成开发环境的窗口，如图 1-2 所示。

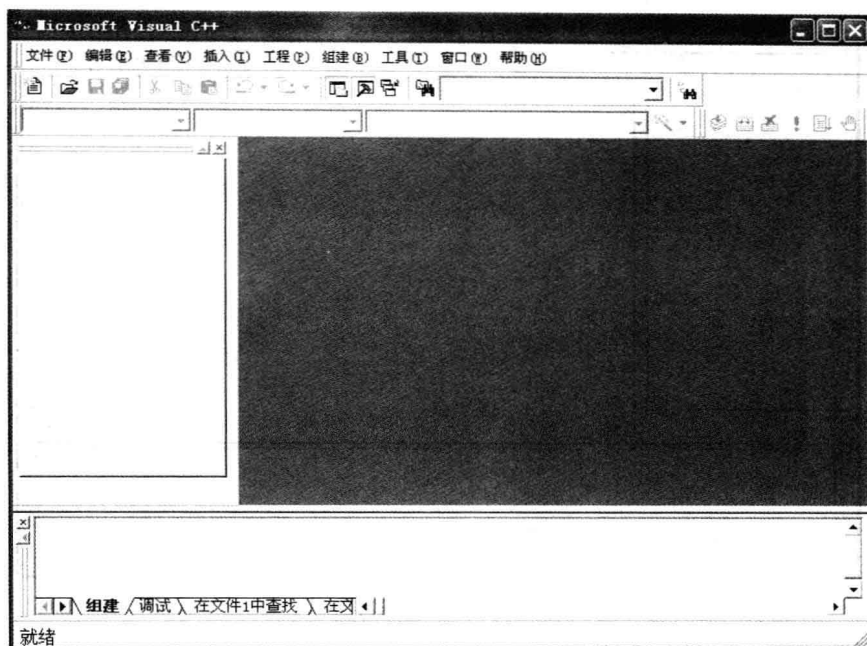


图 1-2 Visual C++ 6.0 集成开发环境的窗口

3) “文件” → “新建” → “文件” 选项卡 → “C++ Source File”，如图 1-3 所示。

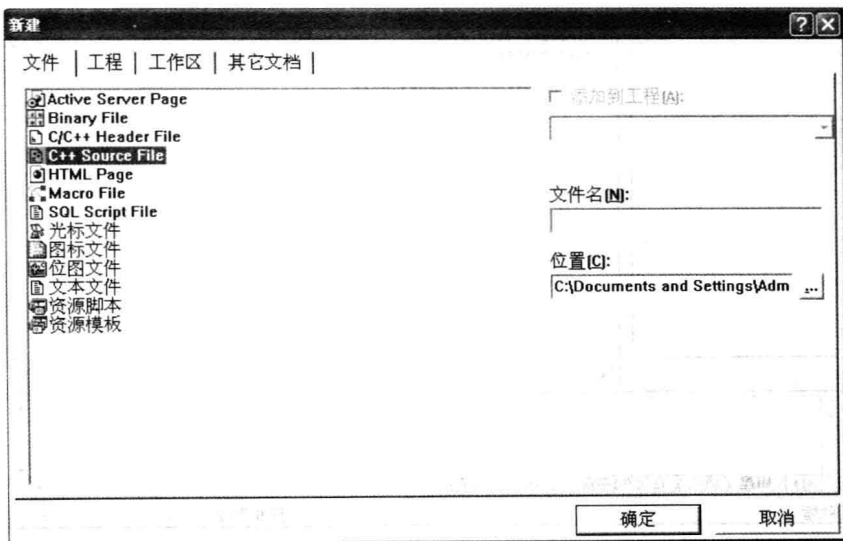


图 1-3 Visual C++ 6.0 “文件” 选项卡

在“文件名”输入框中输入程序名（如：abc），在 Visual C++ 6.0 环境下编辑的 C 语言源程序的扩展名默认是 .cpp。

4) 进入源程序的编辑窗口，如图 1-4 所示。

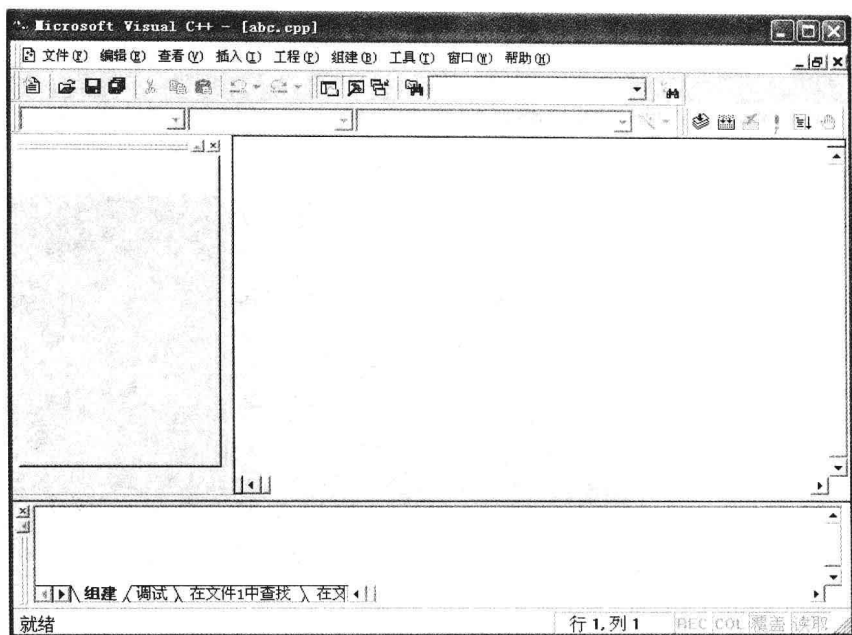


图 1-4 源程序编辑窗口

5) 编写源程序，如图 1-5 所示。

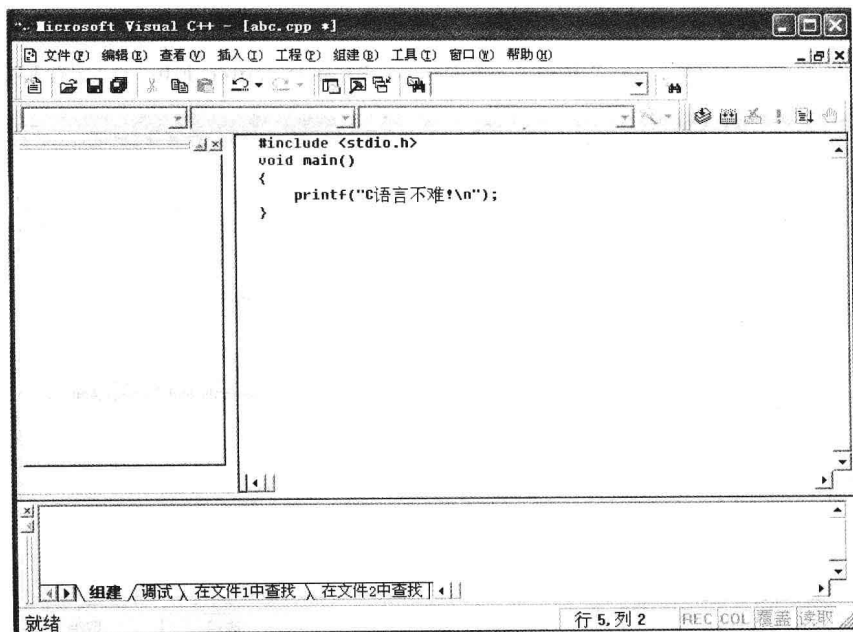


图 1-5 编写源程序

- 6) 编译，如图 1-6 所示。
- 7) 连接，如图 1-7 所示。
- 8) 执行，如图 1-8 所示。
- 9) 显示程序执行结果，如图 1-9 所示。

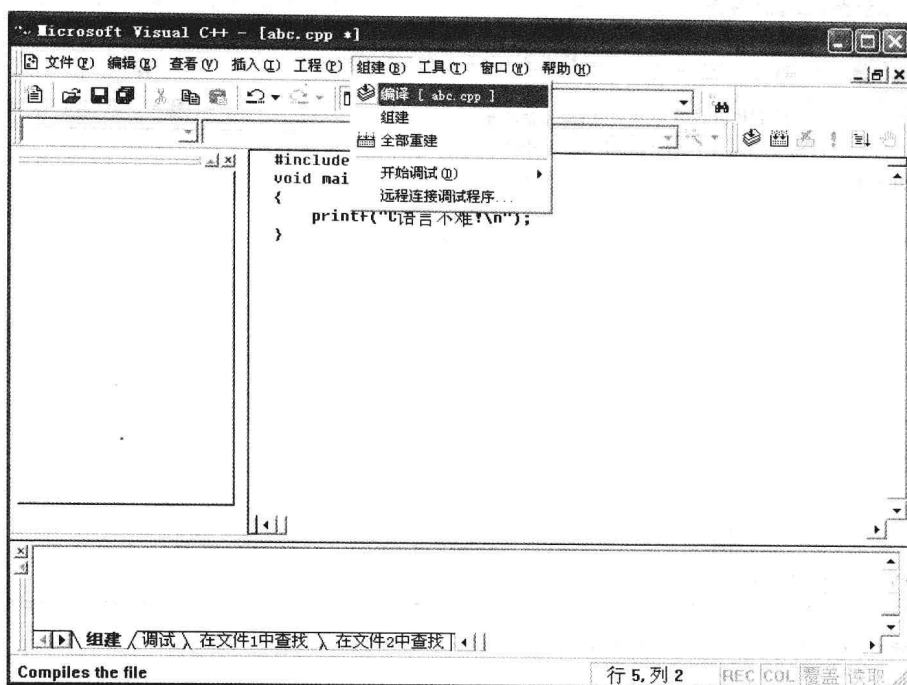


图 1-6 编译

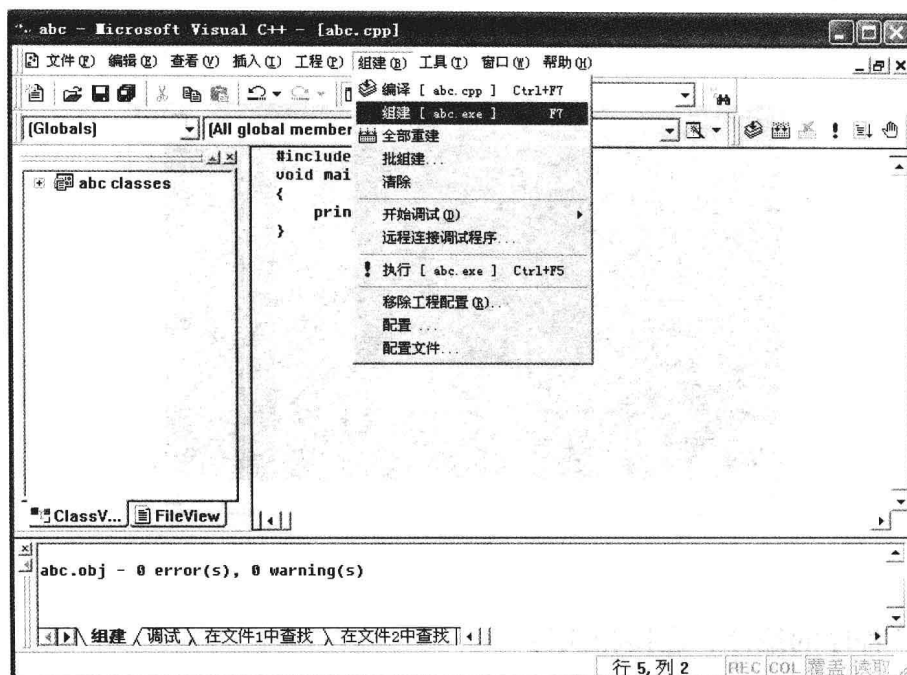


图 1-7 连接

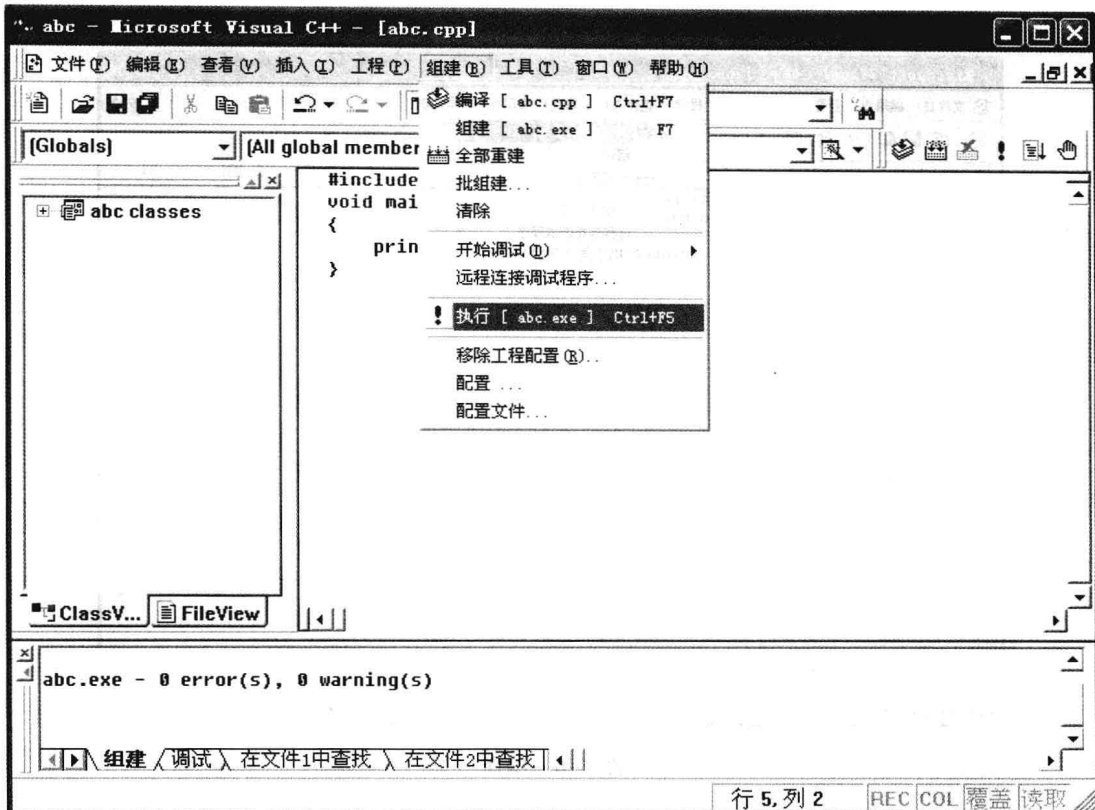


图 1-8 执行

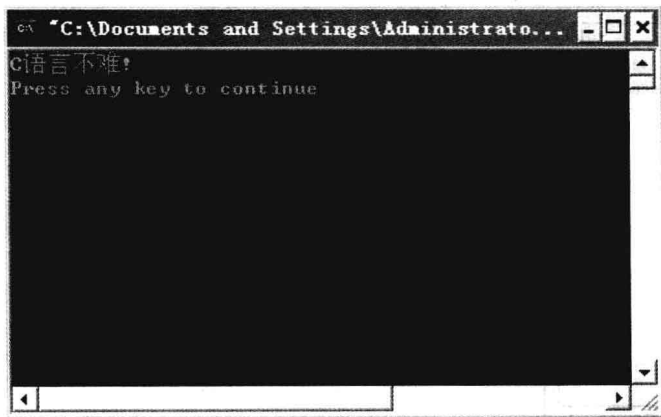


图 1-9 执行结果

1.3.2 C 程序的编译、连接和执行

用计算机程序设计语言编写的程序称为源程序。不同的语言编写的源程序，其扩展名是不同的。在 Visual C++ 6.0 环境中编写的源程序，其扩展名为 .cpp。

源程序是用户根据问题的需要，按照某种计算机语言的编写规则而编写的代码，用户可以识别，而机器不能识别。所以，源程序编好后，还必须使用该语言本身所提供的编译系统，对源程序进行编译，把它翻译成机器可以识别的二进制文件，即目标代码文件，其扩展名为 .obj。目标代码文件产生后，还要连接 C 语言库函数，产生可执行文件，其扩展名为 .exe，可执行文件也是二进制文件。我们最终执行 .exe 文件，显示程序执行结果。

C 程序的编译、连接和执行过程，如图 1-10 所示。

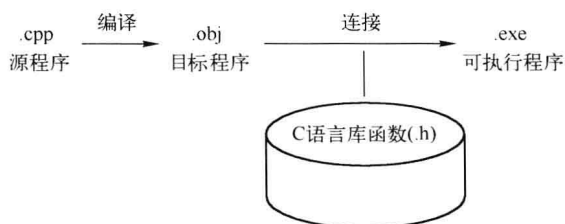


图 1-10 C 程序的编译、连接和执行过程

1.4 简单的 C 程序

【例 1-1】 编写一个最简单的 C 程序。

```
#include <stdio.h>
void main()
{
    printf("C 语言不难! \n");
}
```

运行结果如图 1-11 所示。

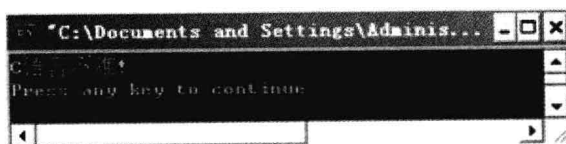


图 1-11 例 1-1 运行结果

说明：

1) C 程序的最简单的结构如下：

```
#include <stdio.h>
void main()
{
    ...;
    ...;
    ...;
}
```

① #include <stdio.h>

所有 C 语言程序的开始都有这一行。这是一个头文件 (.h)，在第 8 章还要详细讲解，这里就知道必须加上这一行就可以了，因为程序中用到有关输入/输出的库函数。在本例中，如果不加这一行，在编译的时候就不能识别 printf() 这个函数了。

② main()

```
{  
    ...;  
    ...;  
    ...;  
}
```

main() 是 C 程序的主函数，每个 C 程序有且只有一个主函数，C 语言程序都是从主函数开始执行的。main 后面的小括号 “()” 是函数的表示形式，其中可以有参数，也可以没有参数，即使没有参数，也必须写上这对小括号 “()”。main 下面的一对大括号 “{...}” 是函数体，即该函数所要完成的事情，根据程序功能的要求有一句写一句，有多少句写多少句，且每一句都是用 “;” 结束。

③ void

表示该函数没有返回值。对我们目前来说，绝大多数情况下主函数都是不需要返回值的。

2) printf (" C 语言不难! \n");

printf() 是 C 语言的输出函数，其功能是显示信息，后面加 “;” 就构成了函数调用语句。该函数小括号 “()” 内，是用一对双引号 “” 引起来的字符串，作为此函数的参数。本例中，字符串包含的信息是：① 需要显示的字符；② 换行符 “\n”，它起着分行的作用。

3) C 语言默认在英文、小写、半角的方式下编写程序。

【例 1-2】 已知两整数 4、5，求它们的和。

```
#include <stdio.h>  
void main() /* 求两整数之和 */  
{  
    int num1,num2,sum; //定义三个变量  
    num1 =4;num2 =5;  
    sum = num1 + num2;  
    printf("两数之和是:%d\n",sum);  
}
```

运行结果如图 1-12 所示。

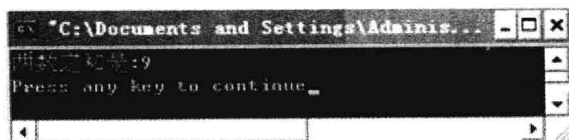


图 1-12 例 1-2 运行结果

说明:

1) `int num1, num2, sum;`

① `num1`、`num2`、`sum` 都是变量的名称。变量是用来存放值的, 因为有“加数”“被加数”“和”3 个值需要存放, 所以定义了3 个变量。

② `int` 表示定义的3 个变量都是整数, 有关数据类型的知识在第2 章还要详细学习。

③ 标识符就是名称。C 语言规定用户定义的标识符只能由字母、数字、下画线3 种字符组成, 且第一个字符必须是字母或下画线。例如, `sum`、`_sys`、`J2_7` 都是合法的标识符; `a-b`、`123a`、`#abc` 都是不合法的标识符。

大写字母和小写字母被认为是两个不同的字符。因此, `sum` 和 `SUM` 被认为是两个不同的变量名。通常, 变量名用小写字母表示。

标识符除了变量名以外, 还有符号常量名、数组名、函数名等, 在后续章节中都会学到。

用户给变量命名的时候有一个原则: 就是“见名知意”, 以增加程序的可读性。

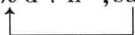
关键字又称为保留字, 是 C 语言规定的具有特定含义的标识符, 用户不能用关键字作为标识符, 如 `printf`、`int` 等。C 语言共有 32 个关键字, 详见附录 IV。

2) `num1 = 4; num2 = 5;`

`sum = num1 + num2;`

在 C 语言中, 符号“=”不是等于, 而是赋值运算符。数学中的等于“=”是从左向右进行运算(如: `a = 3`, 读做 `a` 等于3; 再如: `a + b = 8`, 读做 `a` 与 `b` 之和等于8); 而赋值运算符“=”是从右向左进行运算, 它的意思是将赋值运算符“=”右边的值存放到左边变量所对应的存储单元中(如: `num1 = 4;`, 读做将4 存放到 `num1` 变量中; 再如: `sum = num1 + num2;`, 读做将 `num1` 变量中的值与 `num2` 变量中的值相加, 然后将“和”存放到 `sum` 变量中)。

3) `printf(" 两数之和是: %d \n", sum);`



对于 `printf()` 这个输出函数, 在例 1-1 中已做了简单的介绍, 此输出函数的完整形式如下:

`printf("格式控制", 输出表列);`

通常, 小括号内有两部分参数, 中间用“,”分隔, 左边的参数是用双引号“”引起来的字符串, 右边的参数是输出表列。

左边的参数——“格式控制”包含以下3 种信息。

① 普通字符: 即原样输出的字符(如: 两数之和是:)。

② 格式说明: 它由“%”和格式字符组成(如: `%d`, 表示在此要显示一个整数, 有更多的格式字符在第3 章还要详细学习)。

③ 转义字符: 有特殊意义的字符(如: `\n`)。

右边的参数——输出表列, 是需要输出的一组数据, 可以是变量、表达式、函数等(如: `sum`, 表示将 `sum` 变量中的整数在前面“`%d`”的位置上显示出来)。

4) “,” 和 “;”

C 语言规定, 每个数据定义和语句的最后必须有一个“;”, 各变量之间用“,”分隔。

```

int num1,num2,sum;           //数据定义(此例是变量定义)
num1 =4;                     //语句
num2 =5;                     //语句
sum = num1 + num2;           //语句
printf("两数之和是:%d\n",sum); //函数调用语句

```

5) 注释

```

/* ... */      块注释
//...         行注释

```

注释是对程序、语句的说明，以便阅读。注释在编译时，是不翻译成二进制码的。

6) 书写格式

C 程序书写格式自由，一行可以写几个语句，一个语句也可以写在多行上。

【例 1-3】 求任意两整数之和。

```

#include <stdio.h>
void main()                /* 求任意两整数之和 */
{
    int num1,num2,sum;
    scanf("%d,%d",&num1,&num2);
    sum = num1 + num2;
    printf("两数之和是:%d\n",sum);
}

```

运行结果如图 1-13 ~ 图 1-15 所示。

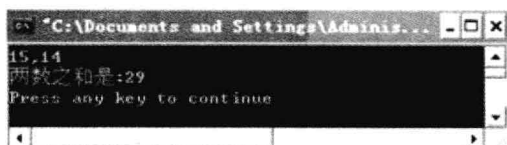


图 1-13 例 1-3 运行结果

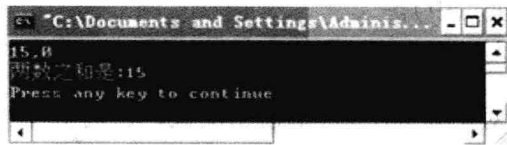


图 1-14 例 1-3 运行结果

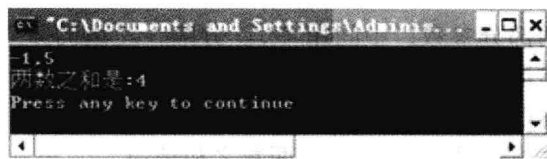


图 1-15 例 1-3 运行结果

说明:

```

scanf( "%d,%d",&num1, &num2 );

```

scanf() 是输入函数，其功能是从键盘上输入值送到变量所对应的存储单元中，后面加“;”就构成了函数调用语句。此输入函数的完整形式如下。