



高等学校计算机类专业规划教材

C 程序设计与实例教程

曾令明 李 飞 编著
甘 刚 宋 波



西安电子科技大学出版社
<http://www.xduph.com>



21 世纪中国计算机教育系列教材

C 程序设计 with 实例教程

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社



清华大学出版社
清华大学出版社

TP312/2564

2007

面向 21 世纪高等学校计算机类专业规划教材

C 程序设计与实例教程

曾令明 李 飞

编著

甘 刚 宋 波

西安电子科技大学出版社

2007

内 容 简 介

C 语言是现今使用最广泛的计算机语言之一,也是学习其他高级语言的基础。本书从 C 语言程序设计的基本原理及基本思想出发,紧扣基础,面向应用。全书共九章,内容包括: C 语言的基本概念、C 语言的语法规则以及利用 C 语言进行程序设计的有关知识。每个知识点均配有完整的程序示范,有利于读者理解和掌握。还针对每章的内容精心设计了大量习题,且书末附有习题参考答案。

本书内容新颖、体系合理、逻辑性强、结构紧凑、文字流畅。适合作为高等学校 C 程序设计课程的教材,也适合广大计算机应用技术人员与计算机等级考试人员学习参考。

图书在版编目(CIP)数据

C 程序设计 with 实例教程 / 曾令明等编著. —西安: 西安电子科技大学出版社, 2007.8

面向 21 世纪高等学校计算机类专业规划教材

ISBN 978-7-5606-1857-9

I. C… II. 曾… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 091003 号

策 划 李惠萍

责任编辑 任 婧 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西光大印务有限责任公司

版 次 2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 15.875

字 数 371 千字

印 数 1~4000 册

定 价 21.00 元

ISBN 978-7-5606-1857-9 / TP · 0964

XDUP 2149001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

C 语言既是目前国内外广泛使用的计算机程序设计语言,也是软件开发人员必须掌握的一种语言。由于 C 语言具有语法结构简洁精巧,目标代码高效,便于描述算法及方便深入系统底层等特点,使得大多数操作系统都采用 C 语言进行开发,如 Windows、UNIX、Linux、Mac、OS/2 等;同时,还有很多新型的程序设计语言也都是 C 语言所衍生的,如 C++、Java、C#、J#、Perl 等。目前,市面上大部分讲述算法和程序设计原理的书都是用 C 语言描述的,因此,只有学会了 C 语言,才能进一步学习这些内容;只要掌握了 C 语言,就可以轻松地学会多门语言,并且可以用这些新型的语言去开发程序。

C 语言不但为计算机专业人员所使用,而且也为广大非计算机专业人员所喜爱和使用,所以许多高等院校不仅在计算机专业开设了 C 语言课程,在非计算机专业同样也开设了 C 语言课程。全国计算机等级考试、全国计算机应用技术证书考试(NIT)和全国各地区组织的大学生计算机统一考试都将 C 语言列入了考试范围,并且目前社会上在招聘程序设计人员时,也将 C 语言作为最基本的开发工具进行考核。因此,学习 C 语言已经成为计算机应用人员和高等院校学生的迫切需求。

本书系统地讲解了 C 语言程序设计基础知识、基本结构程序设计、数组的应用、函数与预处理命令、指针的应用、结构体与共用体、位运算与位段、文件及相关函数等知识。考虑到学生的实际需要,本书在内容安排上力求浅显易懂,以丰富的实例来帮助学生理解所学知识,每章末均配有精心设计的习题,使学生能够及时检查对各章知识点的掌握程度,以促进学生对知识的灵活应用。

全书内容新颖、体系合理、逻辑性强、结构紧凑、文字流畅。适合作为高等院校 C 程序设计课程的教材,也适合广大计算机应用技术人员与计算机等级考试人员学习参考。

限于编者水平,疏漏之处在所难免,恳请广大读者批评指正。

编 者
2007 年 6 月

目 录

第1章 概述.....	1
1.1 学习C语言的必要性.....	1
1.2 C语言简介.....	2
1.2.1 C语言的产生及发展.....	2
1.2.2 C语言的特点.....	2
1.2.3 C程序的组成及书写格式.....	3
1.3 C程序的运行过程.....	4
1.3.1 启动 Visual C++ 6.0.....	5
1.3.2 创建、编辑C程序.....	5
1.3.3 编译、连接、执行C程序.....	7
1.4 练习题.....	8
第2章 数据类型、运算符与表达式.....	9
2.1 关键字与标识符.....	9
2.1.1 关键字.....	9
2.1.2 标识符.....	9
2.2 数据类型.....	10
2.2.1 C语言的数据类型.....	10
2.2.2 C语言的基本数据类型.....	11
2.3 常量.....	12
2.3.1 整型常量.....	12
2.3.2 实型常量.....	13
2.3.3 字符型常量.....	14
2.3.4 符号常量.....	16
2.4 变量.....	17
2.4.1 变量的定义.....	18
2.4.2 变量的基本操作.....	18
2.4.3 整型变量.....	18
2.4.4 实型变量.....	19
2.4.5 字符变量.....	19
2.5 运算符与表达式.....	21
2.5.1 运算符概述.....	21
2.5.2 算术运算符与算术表达式.....	21
2.5.3 赋值运算符与赋值表达式.....	23
2.5.4 表达式中的类型转换.....	23

2.5.5 逗号运算符与逗号表达式	25
2.6 练习题	26
第3章 基本结构程序设计	29
3.1 算法	29
3.1.1 算法的概念	29
3.1.2 算法的描述	31
3.2 顺序结构	35
3.2.1 语句	35
3.2.2 数据的输入和输出	37
3.3 选择结构	44
3.3.1 关系运算与逻辑运算	44
3.3.2 if 语句	47
3.3.3 switch 语句	51
3.4 循环结构	53
3.4.1 while 语句	53
3.4.2 do-while 语句	55
3.4.3 for 语句	57
3.4.4 循环的嵌套	60
3.4.5 break 语句与 continue 语句	62
3.5 练习题	64
第4章 数组	71
4.1 数组概述	71
4.1.1 数组的引入	71
4.1.2 数组的概念	71
4.2 一维数组	72
4.2.1 一维数组的定义	72
4.2.2 一维数组的引用	73
4.2.3 一维数组的初始化	74
4.2.4 一维数组应用举例	75
4.3 二维数组	76
4.3.1 二维数组的定义	76
4.3.2 二维数组的引用	77
4.3.3 二维数组的初始化	78
4.3.4 二维数组应用举例	79
4.4 字符数组	80
4.4.1 字符数组的定义	81
4.4.2 字符数组的初始化	81
4.4.3 字符串	81
4.4.4 字符数组的引用	82

4.4.5 字符数组的输入和输出	82
4.4.6 字符串处理函数	83
4.4.7 字符数组应用举例	87
4.5 练习题	88
第5章 函数与预处理命令	93
5.1 函数的概念	93
5.1.1 模块与函数	93
5.1.2 函数分类	94
5.2 函数的定义与调用	95
5.2.1 函数定义	95
5.2.2 函数调用	96
5.3 函数的嵌套与递归	103
5.3.1 函数的嵌套调用	103
5.3.2 函数的递归	104
5.4 数组作为函数参数	108
5.4.1 数组元素作函数实参	108
5.4.2 数组名作函数参数	109
5.5 变量的作用域	111
5.5.1 局部变量	111
5.5.2 全局变量	113
5.6 变量的生存期	116
5.6.1 静态存储变量	116
5.6.2 动态存储变量	117
5.7 预处理命令	120
5.7.1 宏定义	120
5.7.2 文件包含	128
5.8 练习题	129
第6章 指针	134
6.1 地址和指针的概念	134
6.2 指针变量的定义与使用	135
6.2.1 指针变量的定义	135
6.2.2 指针变量的使用	135
6.3 指针与函数	138
6.3.1 指针作函数的参数	138
6.3.2 返回值为指针的函数	140
6.3.3 指向函数的指针	141
6.4 指针与数组	142
6.4.1 指向一维数组的指针	143
6.4.2 指向多维数组的指针	150

6.4.3 指针与字符串	155
6.4.4 指针数组和指向指针的指针	159
6.5 练习题	164
第7章 结构体与共用体	168
7.1 结构体概述	168
7.1.1 结构体的引入	168
7.1.2 结构体类型的定义	168
7.2 结构体变量	170
7.2.1 结构体变量的定义	170
7.2.2 结构体变量的引用	172
7.2.3 结构体变量的初始化	174
7.2.4 结构体变量作函数参数	175
7.3 结构体数组	176
7.3.1 结构体数组的定义	176
7.3.2 结构体数组的初始化	176
7.3.3 结构体数组元素的引用	177
7.3.4 结构体数组应用举例	177
7.4 结构体与指针	178
7.4.1 指向结构体变量的指针	178
7.4.2 链表	180
7.4.3 指向结构体数组的指针	187
7.4.4 用指向结构体的指针作函数参数	188
7.5 共用体	189
7.5.1 共用体类型的定义	189
7.5.2 共用体变量的定义	190
7.5.3 共用体变量的引用	190
7.6 枚举类型	191
7.7 用 typedef 定义类型	193
7.8 练习题	194
第8章 位运算与位段	199
8.1 位运算	199
8.1.1 按位与运算	199
8.1.2 按位或运算	200
8.1.3 按位异或运算	200
8.1.4 求反运算	200
8.1.5 左移运算	200
8.1.6 右移运算	200
8.1.7 位运算赋值运算符	201
8.2 位段	201

8.3 练习题	202
第9章 文件	204
9.1 文件概述	204
9.1.1 文件的定义	204
9.1.2 文件的分类	204
9.2 文件指针	206
9.3 文件的打开与关闭	207
9.3.1 文件打开函数(fopen)	207
9.3.2 文件关闭函数(fclose)	208
9.4 文件的读/写	209
9.4.1 字符读/写函数 fgetc 和 fputc	209
9.4.2 字符串读/写函数 fgets 和 fputs	212
9.4.3 数据块读/写函数 fread 和 fwrite	214
9.4.4 格式化读/写函数 fscanf 和 fprintf	216
9.5 文件的定位	218
9.5.1 fseek 函数	218
9.5.2 rewind 函数	219
9.5.3 ftell 函数	220
9.6 文件检测函数	221
9.6.1 feof 函数	221
9.6.2 ferror 函数	221
9.6.3 clearerr 函数	221
9.7 练习题	221
附录	224
附录 A 常用字符与 ASCII 代码对照表	224
附录 B 运算符的优先级和结合性	225
附录 C C 语言常用库函数	226
参考答案	232

第1章 概 述

本章首先概括了学习 C 语言的重要性,接着讨论了 C 语言的特点,最后展示了 C 语言程序的组成及上机步骤。需要说明的是,有些概念要随着对后续各章的深入学习才会有深刻的理解,读者不必在一时不懂的问题上停滞不前。

学习重点:

- ▣ 程序设计语言的分类
- ▣ C 语言的特点
- ▣ C 语言程序的组成

1.1 学习 C 语言的必要性

在面向对象思想盛行的时代,在已经拥有了 C++、Java、C# 这些高级语言的今天,为什么还要学习面向过程的 C 语言呢?下面将对此加以说明。

- (1) C 语言是高级语言中的低级语言,低级的编程语言可以让用户更好地了解计算机。
- (2) 相比用其他语言编写的程序,C 语言能用更少的代码实现相同的功能,而运行效率却更高。有时,一个程序所需要的运行速度,只有 C 语言能达到。
- (3) 如果学好了 C 语言,也就能学习现在任何的高级编程语言。因为所有的高级语言都是以 C 语言为基础的(像 Java、C++、C# 等等)。
- (4) 大部分讲述算法和程序设计原理的书都是用 C 语言描述的。只有学了 C 语言才能学好这些课程。
- (5) C 语言已经存在了很多年,它有着广泛的应用基础和大量的可用代码。这就使用户能在现成代码的基础上,快速、高效地编写新的算法和函数。
- (6) UNIX、Linux 的应用越来越广泛,而这两种操作系统都是用 C 语言编写的。
- (7) C 语言是唯一一个向用户阐述指针本质的语言,它使 C 语言变得更加强大。
- (8) 在从事编程开发方面的工作时,C 语言仍然是最普遍需要的语言,所以它值得用户花时间去学。
- (9) 任何内部有微处理器的设备都支持 C 语言。例如微波炉、手机等智能型家电,都是由 C 语言技术来推动的。

1.2 C 语言简介

1.2.1 C 语言的产生及发展

C 语言是由美国人 D.M.Ritchie 于 1972 年在 B 语言的基础上开发出来的, 主要用于编写 UNIX 操作系统。后经多次改进, 到 1977 年出现了可移植的 C 语言编译程序, 这使得用 C 语言编写的 UNIX 系统可在各种计算机上使用。随着 UNIX 的广泛应用, C 语言也得到了普及和推广, 并最终独立于 UNIX 而成为世界上应用最广泛的计算机语言之一。

在 C 语言的推广过程中出现了多种版本, 这些版本虽然类似, 但通常是不兼容的。为了解决这些版本的兼容性问题, 20 世纪 80 年代, 美国国家标准化协会(ANSI)制定了 C 语言标准, 称为 ANSI C。目前流行的各种版本的 C 语言都是以它为基础的, 如 Turbo C、Microsoft C、Quick C 等。虽然这些版本的 C 语言基本部分相同, 但也是有差异的, 读者可以参考相关手册。本书的叙述以 ANSI C 为基础, 所列程序均是在 Turbo C 或 VC++ 6.0 上调试通过的。

1.2.2 C 语言的特点

C 语言之所以能够迅速发展成为备受欢迎的语言之一, 主要是因为它所具有的强大功能。许多著名的系统软件, 如 DBASE IV、UNIX 都是用 C 语言编写的。归纳起来, C 语言具有以下特点。

1. 高级语言中的低级语言

C 语言把高级语言的基本结构和语句与汇编语言的功能结合了起来, 可以像汇编语言一样对位、字节和地址进行操作。C 语言的这种特性使其成为程序设计语言中的全能冠军, 既可以用它来设计应用软件, 也可以用它来设计系统软件。

2. 结构化语言

结构化语言的显著特征是代码和数据的分离, 能够把执行某个特殊任务的代码和数据从程序中分离出来, 使整个程序由一个个模块组成。用结构化语言编写的程序层次清晰, 便于使用、维护以及调试。

用 C 语言编写的程序由函数组成, 每个函数完成一个特定的功能, 由函数作为模块来实现程序的模块化。同时 C 语言具有多种循环、条件语句控制程序流向, 从而使程序完全结构化。

3. 功能齐全

C 语言不仅有丰富的运算符, 还可以把括号、赋值、强制类型转换等作为运算符处理。C 语言具有各种各样的数据类型, 引入了指针概念, 具有构造新的数据类型的机制。另外, C 语言也具有强大的图形功能, 支持多种显示器和驱动器。

4. 简洁、灵活

C 语言仅有 32 个具有特定意义的关键字, 9 种控制语句, 程序书写上也去掉了一切不

必要的成分,因此语句很简洁。与其他语言相比,编译程序对语法检查比较宽松,程序员使用起来就更灵活,写出的程序效率也就更高。

1.2.3 C 程序的组成及书写格式

我们先分析下面两个用 C 语言编写的简单源程序(简称 C 程序),然后总结出 C 程序的组成及书写格式。

【例 1-1】 在屏幕上显示一行文字:“This is a simple C program.”。

完成该任务的 C 程序如下:

```
/*例 1-1*/
#include <stdio.h>
main ()
{
    printf("This is a simple C program.\n");
}
```

这个程序很简单,但它是一个完整的 C 程序。第 1 行是注释,说明这是例 1-1,它不参与程序的编译和执行,只是便于阅读。第 2 行是 C 预处理程序的一条指令,这条指令通知预处理程序把标准输入/输出头文件(stdio.h)中的内容包括到程序中来(关于预处理和头文件在后续章节有介绍,这里读者可以简单地把头文件看成许多函数的集合,在程序中用到了这些函数中的一个或多个)。第 3 行中的 main()表示程序中的“主函数”,C 程序的执行均是由执行 main 函数开始的。第 4 行与第 6 行表示 main 的函数体范围,左花括号“{”表示函数体的开始,“}”表示函数体的结束。第 5 行是 main()的函数体,此函数体只有一条语句,它调用函数 printf()。printf()函数是编译系统提供的、包含在头文件 stdio.h 中的标准库函数,功能是将双引号内的字符串输出到屏幕上,“\n”表示输出后换行。

【例 1-2】 由键盘输入两个整数,然后在屏幕上输出其中的较大者。

完成该任务的 C 程序如下:

```
#include <stdio.h>
void main()
{
    int x, y, z;
    printf("Please input two numbers:\n");
    scanf("%d %d", &x, &y);
    z = max(x, y);
    printf("maxmum=%d", z);
}
int max(int a, int b)
{
    int c;
    if (a>b)
        c = a;
```

```
else  
    c = b;  
return c;  
}
```

该程序除了主函数 `main` 和函数 `printf` 外, 还增加了函数 `scanf` 和 `max`。其中, `scanf` 函数同 `printf` 一样都是由编译系统提供的库函数, 它包含在标准输入/输出头文件(`stdio.h`)中; `max` 函数是用户自定义的函数, 它的作用是比较两个整数的大小, 并返回较大的那个整数。在该程序中, 定义并使用了变量 `x`、`y`、`z`、`a`、`b` 和 `c`(关于变量类型及定义详见第 2 章); 使用了赋值语句, 如 “`c = a`”, 它将等号右边的数值赋给左边; 还使用了结构化控制语句 `if-else`(详见第 3 章), 以增加程序的判断功能。除此之外, 程序中输出的提示信息 “Please input two numbers:” 和 “maxmum=” 更是增加了人机交互性。

通过上面的例子可以总结出 C 程序的组成和书写格式:

(1) C 程序由函数组成, 函数是 C 程序的基本单位。每个程序有且仅有一个主函数 `main()`。可以调用标准库函数, 也可以定义和调用自定义函数。当要调用标准库函数时, 需要在程序前面用预处理命令包含相应的头文件。

(2) 程序的执行均是由执行 `main` 函数开始的, 最后又回到 `main` 函数结束程序。中间可能因调用其他函数而跳出主函数。

(3) 函数由函数首部和函数体两部分组成。函数首部确定函数名并说明函数类型、函数的形式参数名及其类型; 函数体由一对 “{” 和 “}” 括起来的若干语句构成。

(4) 语句可以分为变量定义语句和执行语句。变量定义语句定义变量, 如 “`int x, y, z;`” 和 “`int c;`”; 执行语句完成函数的功能, 如 “`z = max(x, y);`”。每条语句必须有分号作为结束标志。一行可写几条语句, 一条语句也可以分行书写。

(5) C 语言区分大小写字母。一般用小写字母书写程序, 大写字母作常量。

(6) 注释以 “`/*`” 开头, 以 “`*/`” 结尾。注释文字可以是任意字符, 如汉字、拼音、英文等。注释可以放在程序的任何位置, 以提高程序的可读性。它不参与程序的编译和运行, 是可有可无的。

(7) 在书写程序时为了便于阅读、维护和修改, 最好一行一条语句, 遇到嵌套语句向后缩进, 必要时在语句后面加上注释。

1.3 C 程序的运行过程

用高级语言编写的程序称为 “源程序(source program)”。高级语言源程序的执行方式有两种: 一种是 “编译执行”, 另一种是 “解释执行”。编译执行是指源程序需先通过特定的工具软件整体编译、连接成计算机能够识别和执行的语言(称为机器语言), 然后交给计算机执行, 执行时不需要其他应用程序的支持。常见的编译执行的程序语言有 Pascal、C 和 C++。图 1-1 是 C 语言源程序的运行过程。解释执行是指源程序不需要整体编译, 而是读入一句, 解释执行一句, 整个过程需要一定的软件环境来做源代码的解释器。常见的解释执行的程序语言有 Basic 和 Java。

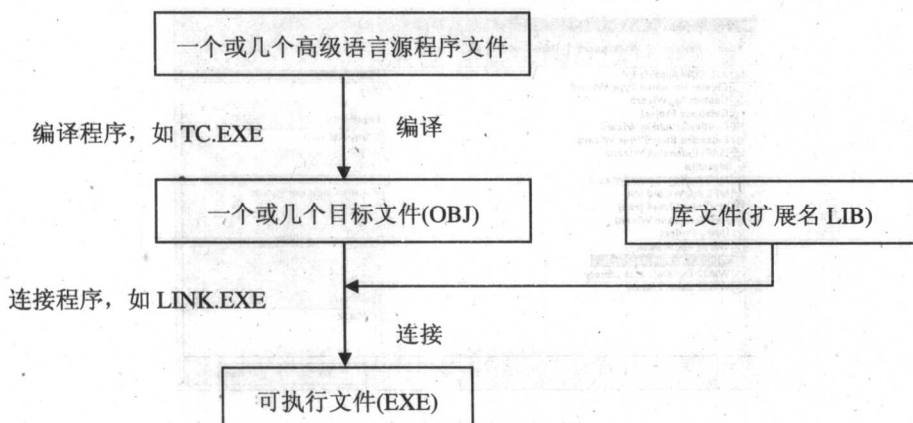


图 1-1 C 语言源程序的运行过程

常用的 C 语言编译系统有 Turbo C 和 Visual C++, 前者是 DOS 环境, 后者是 Windows 环境, 读者可以选择其中一种。这里我们简要介绍用 Visual C++ 6.0 这种比较流行的 C 语言编程环境进行编程的基本步骤, 使读者能够用它进行简单的 C 程序开发。如果读者希望对它有更深入的了解, 请查看其帮助文档和用户手册。

1.3.1 启动 Visual C++ 6.0

在安装了 Visual C++ 6.0 的系统中, 选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”, 即可启动 VC++, 屏幕上将显示如图 1-2 所示的窗口。

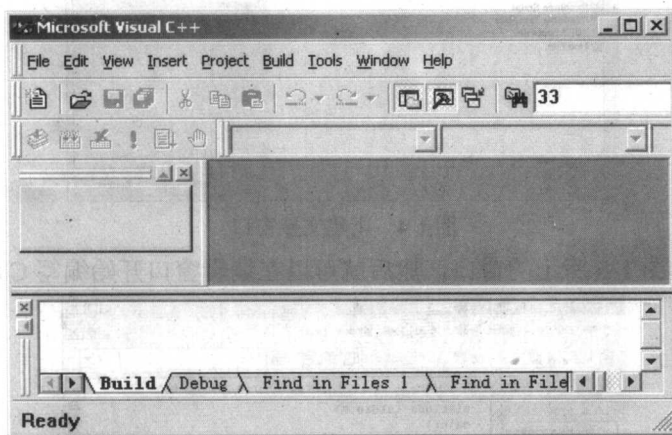


图 1-2 VC++ 启动窗口

1.3.2 创建、编辑 C 程序

启动 Visual C++ 6.0 后, 选择菜单 File 的菜单项 New, 弹出 New 对话框, 如图 1-3 所示。在 New 对话框中选择 Projects 选项卡, 在它的列表框中选择 Win32 Console Application 列表项, 在屏幕右边的 Project name 编辑框中输入项目名称(即程序名称), 如“Exa”, 在 Location 编辑框中指定程序所放位置, 如“E:\myvc”, 然后单击 OK 按钮。

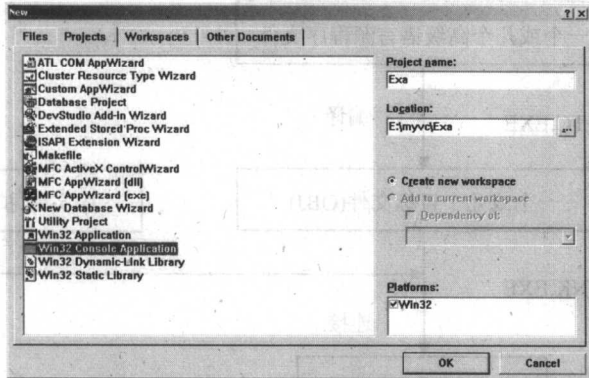


图 1-3 新建项目窗口

接着进入一个 Win32 应用程序向导对话框，选中该对话框中的默认选项 An empty project 选项，单击 Finish 按钮系统将显示 AppWizard 的创建信息，单击 OK 按钮完成项目的创建。然后选择 File 菜单的菜单项 New，在弹出的 New 对话框中选择 Files 选项卡，如图 1-4 所示，在它的列表框中选择 C++ Source File 列表项，在右边的 File 编辑框中输入文件名，如 Exa1，最后单击 OK 按钮。

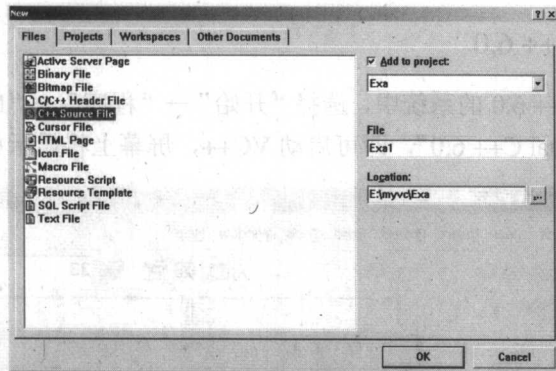


图 1-4 新建文件窗口

屏幕上出现如图 1-5 所示的窗口，然后就可以在编辑窗口开始编写 C 程序。

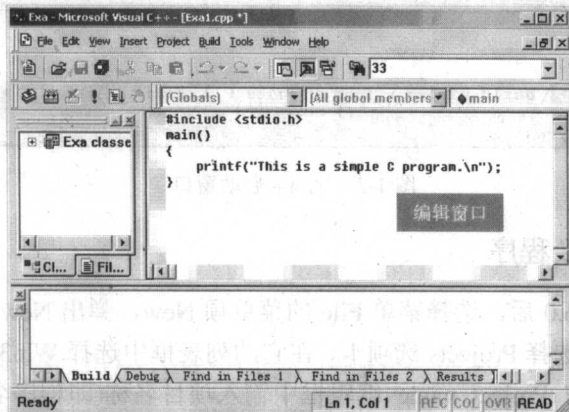


图 1-5 编辑 C 程序窗口

1.3.3 编译、连接、执行 C 程序

在 Visual C++ 6.0 环境下, 源程序的编译、连接和执行可一起完成, 方法是单击工具栏上的“!”按钮。如果程序没有任何错误, 将在图 1-6 所示的信息窗口中显示内容:

0 error(s) 0 warning(s)

有时出现的几个警告性信息(warning)并不影响程序执行。程序执行时, VC++ 将自动弹出数据输入/输出窗口, 按任意键将关闭该窗口。

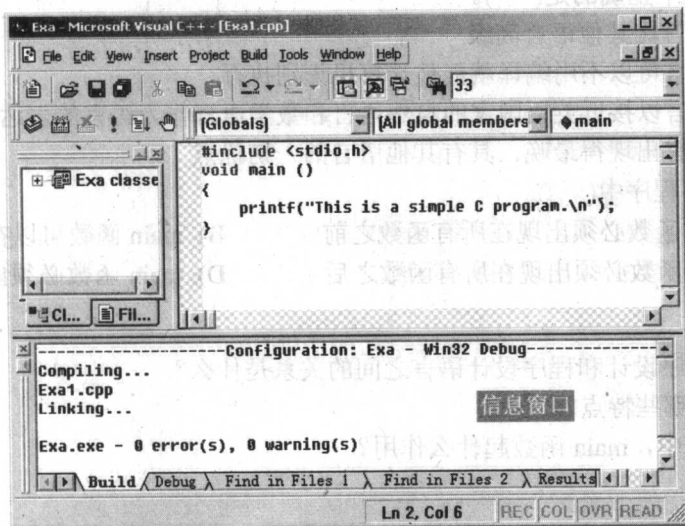


图 1-6 编译正确时的信息窗口

假如程序中有致命性错误(error), 如图 1-7 所示的信息窗口。此时, 双击某行出错信息, 程序窗口中便会指示对应的出错位置, 根据信息窗口的提示分别予以纠正后再编译执行。

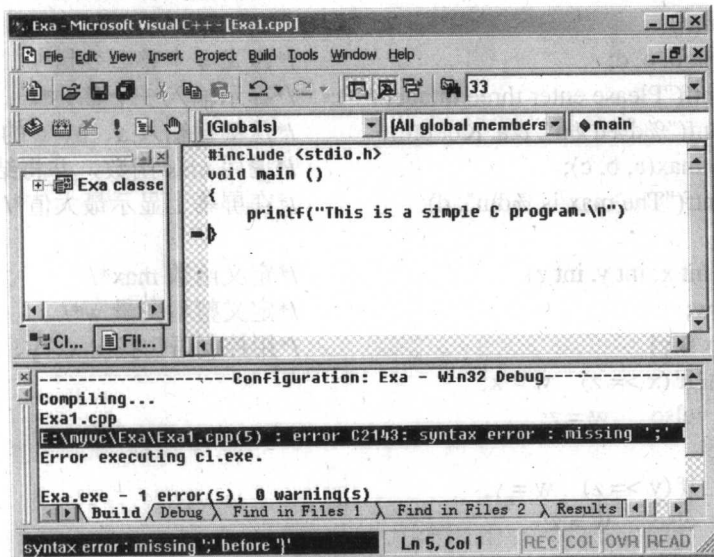


图 1-7 编译出错时的信息窗口