

★ ★ ★
电气信息类
精品系列
★ ★ ★

C语言程序设计

◎主 编 闫会昌 罗勇胜



C 语言程序设计

闫会昌 罗勇胜 主 编

周爱民 副主编

 **北京理工大学出版社**

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

C 语言是国内外广泛使用的程序设计语言，也是计算机专业人员和计算机应用人员学习程序设计的首选语言。

本书共分十章：C 语言概述；数据类型、运算符和表达式；顺序结构设计；选择结构设计；循环结构设计；数组；函数；指针；结构体和联系体；文件。

本书力求使初学者能够建立正确的 C 语言概念，学会基本的编程方法，形成对 C 语言的一个整体了解。本着精讲多练的原则，书中尽量避免用大段文字来叙述概念，而是通过精心设计的图形和大量例子来加以说明。每章后面一般都配有三种练习题：选择题、填空题和编程题，以帮助读者检测 C 语言知识掌握程度，提高程序设计能力。

版权专有 侵权必究

图书在版编目 (CIP) 数据

C 语言程序设计 / 闫会昌，罗勇胜主编. —北京：北京理工大学出版社，2015.5

ISBN 978 - 7 - 5682 - 0486 - 1

I. ①C… II. ①闫… ②罗… III. ①C 语言 - 程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 073631 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(批销中心) 68911084(读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市天利华印刷装订有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 12.25

字 数 / 280 千字

版 次 / 2015 年 5 月第 1 版 2015 年 5 月第 1 次印刷

责任校对 / 陈玉梅

定 价 / 45.00 元

责任印制 / 边心超

图书出现印装质量问题，本社负责调换

前 言

C 语言是国内外广泛使用的程序设计语言之一。C 语言具有表达能力强，语言简洁、紧凑，使用方便、灵活，易学易用等特点。它的数据类型丰富，具有现代语言的各种数据类型。C 语言生成的代码质量高，其代码效率仅比用汇编语言写的代码低 10~20%，并且其可移植性好，有着自己独特的魅力。使用 C 语言进行程序设计已成为软件开发的一个主流，是计算机及相关专业人员学习计算机程序设计的首选语言。

针对学生的特点，本书对 C 语言不常用的细节或非基本的内容作了删减（如输入输出的详细控制），但不失完整性，力求把重点集中在程序设计上。书中所选例题丰富、实用，在讲解基本编程的基础上，努力开拓学生的思路。

本书作为程序设计的入门教材，也适当地介绍了较高级编程部分。本书共分 10 章，包括：第 1 章 C 语言概述；第 2 章数据类型、运算符和表达式；第 3 章顺序结构设计；第 4 章选择结构设计；第 5 章循环结构设计；第 6 章数组；第 7 章函数；第 8 章指针；第 9 章结构体和联合体；第 10 章文件。本书在内容安排上遵循深入浅出、循序渐进的原则，重点是让学生了解程序设计语言的基本知识、掌握程序设计语言的基本方法，使学生具有使用 C 语言解决实际问题的初步能力。各章后面一般都配有三种练习题：选择题、填空题和编程题，有很多都是国家 C 语言二级考试题目，可以帮助读者检测 C 语言知识掌握程度，应对各类考试也有的放矢。

纵观 C 语言教材，大多采用 Turbo C 2.0 开发环境，也有个别介绍 Visual C++ 6.0 环境。由于目前大多都是使用 Windows 操作系统，Turbo C 2.0 不支持鼠标操作也不能使用汉字，在使用上颇感不便；而 Visual C++ 6.0 又过于庞大，学生不能随时随地地在任何电脑上使用。故此，本教材采用 Dev-C++ 集成开发环境，它是一个 Windows 下的 C 和 C++ 程序的集成开发环境，它使用 MingW32/GCC 编译器，遵循 C/C++ 标准。其压缩包只有不到 9.5M，方便携带，是学习 C 或 C++ 很好的开发工具。本书全部例子均在该系统下调试通过。

本书由闫会昌老师担任第一主编（编写第 5、6、7、8 章），罗勇胜老师担任第二主编（编写第 9、10 章，附录 A、B），周爱民老师担任副主编（编写第 1、2、3、4 章），全书由闫会昌老师统稿和主审，并完成程序的调试工作。

编者

目 录

第1章 C语言概述	1
1.1 程序设计语言概述	1
1.1.1 程序设计语言的发展	1
1.1.2 C语言简史及特点	2
1.2 简单的C程序介绍	3
1.2.1 源程序的书写格式	3
1.2.2 C程序结构	4
1.2.3 C语言中的标识符	5
1.3 C程序的编辑、编译、连接和运行	6
1.3.1 C程序的运行步骤	6
1.3.2 Dev-C++多国语言版开发环境简介	6
习题	9
第2章 数据类型、运算符与表达式	11
2.1 C语言的数据类型	11
2.2 常量	12
2.3 变量	15
2.4 变量的初始值	15
2.5 数值型数据间的运算	16
2.6 运算符、表达式及常用数学函数	19
2.6.1 算术运算符及算术表达式	20
2.6.2 常用数学函数	21
2.6.3 赋值运算符及其表达式	22
2.6.4 其他运算符	23
习题	24
第3章 顺序结构设计	27
3.1 结构化程序概述	27
3.2 C语言语句	27
3.3 字符的输入与输出	28
3.3.1 字符输出	28
3.3.2 字符输入	29
3.4 格式数据的输入与输出	29
3.4.1 格式输出	29
3.4.2 格式输入	31

3.5 程序举例	33
习题	35
第 4 章 选择结构程序设计	39
4.1 关系运算符及其表达式	39
4.2 逻辑运算符及其表达式	40
4.3 if 语句	41
4.4 switch 语句	46
4.5 综合应用举例	48
习题	49
第 5 章 循环结构程序设计	53
5.1 while 语句	53
5.2 do-while 语句	57
5.3 for 语句	58
5.4 break 和 continue 语句	60
5.5 多重循环	61
5.6 综合应用举例	63
习题	66
第 6 章 数组	71
6.1 数组概述	71
6.2 一维数组	72
6.2.1 一维数组的定义	72
6.2.2 一维数组的引用	72
6.2.3 一维数组的初始化	73
6.2.4 一维数组的程序举例	73
6.3 二维数组	75
6.3.1 二维数组的定义	75
6.3.2 二维数组的引用	75
6.3.3 二维数组的初始化	76
6.3.4 二维数组的程序举例	77
6.4 字符数组和字符串	78
6.4.1 字符数组与字符串的关系	78
6.4.2 字符数组的输入与输出	79
6.4.3 字符串处理函数	80
6.4.4 字符数组的程序举例	82
6.4.5 字符串数组	83
6.5 综合应用举例	84
习题	88
第 7 章 函数	92
7.1 函数定义	92

7.2 函数参数和函数返回值	93
7.2.1 形式参数和实际参数	93
7.2.2 函数返回值	94
7.3 函数调用	95
7.3.1 函数调用的一般形式	95
7.3.2 函数调用的方式	95
7.3.3 对被调函数的声明	96
7.3.4 函数的嵌套调用	97
7.4 递归函数	98
7.5 数组作函数参数	100
7.5.1 数组元素作函数的实参	100
7.5.2 数组名作函数参数	101
7.6 变量的存储属性	103
7.6.1 动态变量	103
7.6.2 静态变量	104
7.6.3 外部变量	105
7.6.4 内部函数和外部函数	107
7.7 预处理	107
7.7.1 宏定义	107
7.7.2 文件包含	109
7.7.3 条件编译	110
习题	111
第8章 指针	118
8.1 指针的概念、定义和引用	118
8.1.1 指针的概念	118
8.1.2 指针变量的定义	119
8.1.3 指针变量的引用	119
8.2 指针与函数	121
8.2.1 指针变量作函数参数	121
8.2.2 返回值为指针的函数	121
8.2.3 通过指针调用函数	122
8.3 指针与数组	123
8.3.1 指针与数组的关系	123
8.3.2 数组名作函数参数的指针解释	126
8.3.3 指针数组	127
8.3.4 多级指针	128
8.4 指针与字符串	129
8.4.1 字符串的表示形式	129
8.4.2 字符串指针作函数参数	130

8.4.3 返回字符串的函数	132
8.4.4 字符指针数组	132
8.4.5 字符指针与字符数组的讨论	133
8.5 指针数据类型小结	135
8.5.1 各种指针定义小结	135
8.5.2 指针运算小结	135
8.5.3 void 类型指针	136
习题	137
第 9 章 结构体与联合体	141
9.1 结构体定义	141
9.1.1 结构体定义格式	141
9.1.2 结构体变量	143
9.2 结构体数组	145
9.3 结构体与指针	146
9.3.1 结构体指针	146
9.3.2 结构体数组指针	147
9.4 结构体与函数	147
9.4.1 参数为结构体类型	147
9.4.2 函数的返回值为结构体类型	149
9.5 动态内存管理	151
9.5.1 常用动态内存管理函数	151
9.5.2 链表	153
9.6 联合体	156
9.7 枚举类型	157
9.8 用 typedef 定义新类型名	159
习题	160
第 10 章 文件	165
10.1 文件概述	165
10.2 内部文件和外部文件	166
10.3 文件的打开与关闭	166
10.3.1 文件的打开	166
10.3.2 文件的关闭	168
10.4 文件的读/写	168
10.4.1 读/写单个字符	168
10.4.2 读/写字符串	170
10.4.3 文件的格式化读/写	172
10.4.4 块读/写	173
10.5 文件的定位	175
10.5.1 回绕	175

10.5.2 随机定位	176
习题	177
附录 A 常用字符与 ASCII 代码对照表	180
附录 B 常用 C 语言库函数	181
参考文献	184

第1章 C语言概述

C语言是编程语言中较为流行的一种。随着计算机的普及和发展，C语言在各个领域的应用越来越广泛。几乎各类计算机都支持C语言的开发环境，这为C语言的普及及应用奠定了基础。

本章要点：

- (1) 程序设计语言概述；
- (2) 简单的C程序介绍；
- (3) C程序的编辑、编译、连接和运行。

1.1 程序设计语言概述

人与人之间的交流是通过语言完成的。当人们和计算机打交道时，也要涉及语言问题，这就是计算机语言。计算机语言是指一切用于人与计算机之间交换信息的工具，它包括程序设计语言、命令语言、查询语言、定义语言等。程序设计语言也叫编程语言，它是泛指用于书写计算机程序的语言，包括机器语言、汇编语言和高级语言。计算机程序是指人们根据处理问题的需要而事先安排好的需要计算机执行的处理步骤。学习程序设计语言，不但要掌握该程序设计语言的主要规则，还要学习一些程序设计知识，了解程序设计过程，能设计和编写简单的程序。

1.1.1 程序设计语言的发展

1. 机器语言

机器语言是指计算机本身自带的指令系统。计算机的指令由二进制序列组成，用来控制计算机进行某种操作。指令由操作码和地址码两部分组成。其中操作码规定计算机要做的运算；地址码告诉计算机是由哪些数来参加运算，在什么地方能找到数据，计算机的结果存放在哪里，等等。用机器语言编写的程序不必通过任何翻译处理，计算机就能够直接识别和执行。因此，用机器语言编写的程序，具有质量高、执行速度快和占用存储空间少等优点。但是，用机器语言编写的程序无明显特征，难学、难记、难检查和难修改。机器语言属于低级语言。

2. 汇编语言

由于机器语言的缺点，使人们几乎无法直接使用。人们开始把机器指令中的操作码用助词符代替，指令中的地址码也用符号地址来代替，使得机器语言得以“符号化”。汇编语言的指令与机器语言的指令基本上保持了一一对应的关系。与机器语言比起来，汇编语言易记、易读、易检查、易修改，但却不能被计算机直接识别和执行，必须由翻译程序（即汇编程序）翻译成机器语言程序（即目标程序）后才能执行。汇编语言仍然是面向特定计算机的语言，

因此，通用性和可移植性较差。汇编语言仍然属于低级语言。

3. 高级语言

机器语言和汇编语言都是面向特定计算机的语言，虽然执行效率高，但编程的效率却很低。而高级语言则是接近人类自然语言和数学语言的一种程序设计语言，很容易被人们接受和掌握。

高级语言具有以下共同的特点：

- (1) 可以不了解机器语言代码即可编程，是不依赖于特定计算机的语言。
- (2) 高级语言的一个执行语句通常对应多条机器指令。
- (3) 高级语言接近自然语言和数学语言，容易被人们接受和掌握。
- (4) 用高级语言编写的程序（也称“源程序”），可移植性好。
- (5) 源程序必须经过翻译才能被计算机识别和执行。
- (6) 具有结构化程序设计结构（即顺序结构、分支结构和循环结构）和模块化程序设计思想。

翻译的方式有编译和解释两种。编译方式是把源程序从头至尾整个地翻译成目标程序，然后执行该目标程序。解释方式是把源程序逐句翻译，译出一句就立即执行一句。解释方式不产生目标程序。

C 语言就是一种高级语言，它用比较接近人的思维和表达问题方法的形式来描述问题、编写计算机程序，然后以编译的方式进行翻译。

4. 面向对象的程序设计语言



图 1.1 对象示例

面向对象的程序设计语言最突出的特点是封装性，即将那些功能独立的模块和数据封装起来，成为一个个独立的对象。在 Windows 中常见的组合框和命令按钮（如图 1.1 所示）等都是对象。Visual C++、Delphi 等都是具有代表性的面向对象的程序设计语言。

1.1.2 C 语言简史及特点

1. C 语言的发展简史

C 语言是在美国贝尔实验室诞生的。贝尔实验室的 M.D.Ritchie 和 K.Thompson 在开发 UNIX 操作系统时，希望有一种既有高级语言特性，又有低级语言特点的语言来描述系统软件，但当时存在的 CPL、BCPL、B 语言不是规模大、难以实现，就是过于简单，功能很弱。1972—1973 年 M.D.Ritchie 设计出了 C 语言。在一段时间里，C 语言还只是在贝尔实验室内使用，直到 1975 年第 6 版的 C 语言才引起人们的注意。1977 年出现了可移植 C 语言，1978 年第 7 版的 C 语言成了当时的标准，称标准 C。1983 年，美国国家标准化协会（ANSI）对 C 语言制定新的标准，称为 ANSI C，这就是现行的 C 语言标准。

此后，C 语言的发展非常迅速，现在已风靡全世界，成为最受人们欢迎的几种计算机程序设计语言之一。

目前，实用的 C 语言编译系统种类繁多，如 Turbo C，Microsoft C，Quick C 等，虽然它

们略有差异，但基本部分都是相同的。

本书的全部例子采用 Dev-C++ 4.9.9.2 集成开发环境。

2. C语言的特点

- (1) 表达能力强。
- (2) 简洁、紧凑，使用方便、灵活，易学易用。
- (3) 数据类型丰富，具有现代语言的各种数据类型。
- (4) 语言生成的代码质量高，其代码效率仅比用汇编语言写的代码低 10%~20%。
- (5) C 语言的可移植性好。

1.2 简单的 C 程序介绍

1.2.1 源程序的书写格式

例 1.1 Hello 程序

```
main()
{
    printf("Hello!\n");
    system("pause");
}
```

程序运行后在屏幕上显示：

Hello!

请按任意键继续...

说明：

- (1) main()是主函数，相当于其他语言中的主程序。
- (2) 一对大括号是函数的开始与结束，被括起来的部分称为函数体，本函数体内有两个语句。
- (3) printf()是 C 语言的输出库函数，双引号括起来的字符串按原样输出。其中以“\”开始的字符称“转义字符”，“n”是换行符，即在输出“Hello”后回车换行。如果再运行一次该程序，两次的输出会分别在两行上显示出来。
- (4) system(“pause”)是调用操作系统的暂停，将会在屏幕上显示“请按任意键继续...”，目的是方便观察运行结果。
- (5) 每个语句后面都有一个分号。

例 1.2 两个数相加

```
main()
{
    int a,b,sum;                /* 定义 3 个整型变量 a, b 和 sum */
    scanf("%d,%d",&a,&b);        /* 将输入的两个数存入 a 和 b 中 */
    sum=a+b;
    printf("sum=%d\n",sum);
    system("pause");
}
```

程序运行：

```
12,34↵
```

(有下画线的表示从键盘输入，其中"↵"表示按回车键)

```
sum=46
```

请按任意键继续...

说明：

(1) 用“/*”和“*/”括起来的部分是注释内容。注释只是给人看的，对编译和运行不起作用。注释可以加在程序的任何位置，它的使用是为了提高程序的可读性。

(2) `int a,b,sum;` 是定义变量部分，定义了3个 `int`(整型)变量 `a`、`b` 和 `sum`。经定义后，程序为这些变量开辟该类型所需要的内存单元，用来存入一个值。如果在程序中使用变量，就必须先定义。

(3) `scanf()`是输入函数，也是库函数。其作用是将键盘输入的数据存入变量 `a` 和 `b` 中。`&a` 和 `&b` 中“&”的含义是“取地址”，即将两个数值分别存入到变量 `a` 和 `b` 的地址所标志的内存单元中。

(4) `scanf()`和 `printf()`两个函数中以“%”开始的字符是输入输出“格式描述字符”，用来指定输入输出时的数据类型和格式。“%d”表示“十进制整型数类型”。本例中要求输入两个整数存入 `a` 和 `b` 中，输出一个整数 `sum` 的值。

(5) 在 `scanf()`和 `printf()`中双引号中除了转义字符和格式字符以外的其他字符，在输入时要求原样输入，在输出时原样输出。如“%d,%d”中的逗号，输入时要求将其输入，“sum=”输出时它将被输出。

(6) `sum=a+b;`是赋值语句，其作用是将“=”右边表达式的值计算出来，送给左边的变量。

(7) C 程序书写格式自由，允许在一行上写多条语句，也允许把一条语句写在多行上。

1.2.2 C 程序结构

C 语言是基于函数的语言，C 程序由函数组成。函数分为库函数和自定义函数两类。

1. 库函数

库函数（也称系统函数或标准函数）是系统已经编写好的，根据它们的不同功能和性质分门别类地存放在不同的文件中，称之为库文件或头文件，其文件的扩展名为.h。

若在程序中使用库函数，应在程序开始处指明它们在哪个库文件中，这个过程称“文件包含”，由编译预处理命令的`#include`完成。其语法格式为：

```
#include <库文件名> 或 #include "库文件名"
```

2. 用户自定义函数

用户自定义函数是由用户根据需要自己在程序中编写。其语法格式为：

```
类型符 函数名(形式参数说明表)          /* 函数首部 */
{  变量定义部分                          /* 函数体 */
  语句部分
}
```

在函数首部中，“类型符”是指当函数结束时所得到的结果类型。“函数名”是给函数起的名字。“形式参数说明表”是指当函数在执行时所需要的“自变量”，它是作为函数计算的

依据。“形式参数”是变量，要对它进行类型说明，“形式参数”可以有多个，在书写时它们中间用逗号分隔。

函数体是被花括号“{...}”括着的部分。它包括两部分。

(1) 变量定义部分：如例1.3中的“float z;”。

(2) 语句部分（也称执行部分）：如例1.3中的“z=3.14*pow(r,2);”。

例 1.3 使用函数求圆的面积

```
#include <math.h>
float area(float r)           /* 用户自定义函数 */
{
    float z;
    z=3.14*pow(r,2);          /* pow 是库函数，用来计算 r 的平方 */
    return z;
}
main()                        /* 主函数 */
{
    float a,s;
    printf("请输入半径:");
    scanf("%f",&a);
    s=area(a);
    printf("半径是%f 的圆的面积为%f\n",a,s);
    system("pause");
}
```

程序运行：

请输入半径：10✓

半径是 10.000000 圆的面积为 314.000000

请按任意键继续...

说明：

(1) float 是实型（单精度类型）类型符，说明其后面的变量或函数为实型。对于实型数据的输入与输出使用“%f”格式描述字符，输出时有6位小数。

(2) 3.14*pow(r,2)是用来求数学公式 $3.14 \times r^2$ 的 C 语言表达式，即所有的数学公式都必须转换为 C 语言表达式。

(3) 一个 C 程序存储在一个或多个文件中，函数是构成 C 程序的基本程序单位。即 C 程序由一个或多个函数组成，组成程序的若干函数中只能有一个主函数 main()，C 程序总是从主函数开始执行。

(4) 用户自定义函数可以放在主函数之前，也可以放在主函数之后，但总是先执行主函数。用户自定义函数和库函数都是通过主函数直接或间接地调用（使用）。在没有学习函数章节之前，建议将自定义函数写在主函数之前。

归纳起来，C 程序的结构形式如下：

C 程序=编译预处理命令+C 函数 1+C 函数 2+...+C 函数 n

1.2.3 C 语言中的标识符

标识符是用来命名一个名字的，如常量名、变量名、函数名、数组名、类型名等都必须

是 C 语言中合法的标识符。

C 语言规定标识符只能由字母、数字和下划线 3 种字符组成，且第一个字符必须为字母或下划线。例如：

a, x5, count, year, average, student_number, classroom3

/* 都是合法的标识符 */

\$123, x+y, road#2, student name, a=b

/* 都是非法的标识符 */

注意：大写字母和小写字母在 C 语言中被认为是两个不同的字符。因此 count 和 COUNT 是两个不同的标识符。

在起名时，应注意做到“见名知义”。除了代数计算外，一般不要用代数符号（如 a, b, c, x1, y23 等）作为名字，这样可以增加程序的可读性。

1.3 C 程序的编辑、编译、连接和运行

1.3.1 C 程序的运行步骤

C 程序要真正运行需要经过 4 个步骤：

(1) 编辑。将源程序录入到计算机中，并可进行修改。保存时产生以“.C”为扩展名的源程序文件。

(2) 编译。源程序不能直接执行，需要经过 C 编译程序，它将把源程序“翻译”成由机器指令组成的目标程序文件，其扩展名为“.OBJ”。

(3) 连接。目标文件仍不能立即执行，因为，每一个程序模块往往是单独编译形成多个 OBJ 文件，把各个 OBJ 文件以及系统提供的系统库函数等进行连接处理后，才能形成可执行文件，其扩展名为“.EXE”。

(4) 运行。运行可执行文件，得到运行结果。

对源程序(*.C)进行编译如果没有错误，则生成目标文件(.OBJ)；否则不生成，并要求对源程序进行修改后再重新编译，直到编译没有错误再进行连接。连接如果没有错误，则生成可执行文件(.EXE)；否则不生成，并要求对源程序进行修改后再重新编译、连接，直到没有错误时方可运行程序。程序运行后得到的结果如果不正确，同样要修改源程序。以上过程如图 1.2 所示。

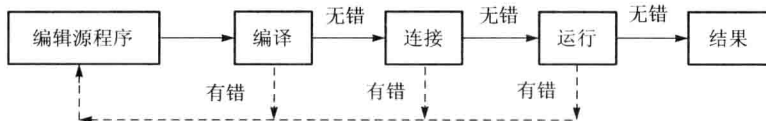


图 1.2 C 程序的运行步骤

1.3.2 Dev-C++多国语言版开发环境简介

Dev-C++是一个 Windows 下的 C 和 C++程序的集成开发环境。它使用 MingW32/GCC 编译器，遵循 C/C++标准。开发环境包括多页面窗口、工程编辑器以及调试器等，在工程编辑器中集合了编辑器、编译器、连接程序和执行程序，提供高亮度语法显示的，以减少编辑错

误，还有完善的调试功能，能够满足初学者与编程高手的不同需求，是学习 C 或 C++ 的首选开发工具。多国语言版中包含简繁体中文语言界面及技巧提示，还有英语、俄语、法语、德语、意大利语等二十多个国家和地区语言提供选择。

启动 Dev-C++ 后，显示 Dev-C++ 的集成开发环境窗口，如图 1.3 所示。

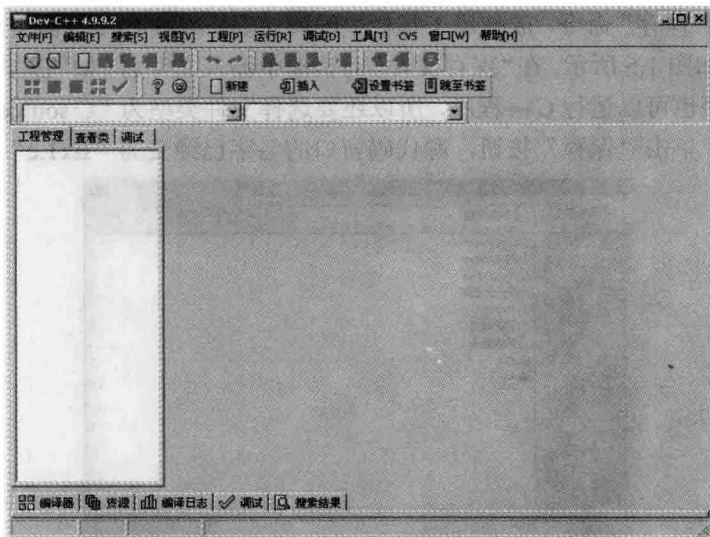


图 1.3 Dev-C++ 的集成开发环境窗口

1. 建立 C 语言源程序

在图 1.3 的集成开发环境中，选择“文件”|“新建”|“源代码”命令，或使用工具栏中的  工具，或直接按 Ctrl+N 组合键，都会在集成开发环境的右侧出现源代码编辑窗口。初始时，源代码文件名默认是“未命名 1”，如图 1.4 所示。

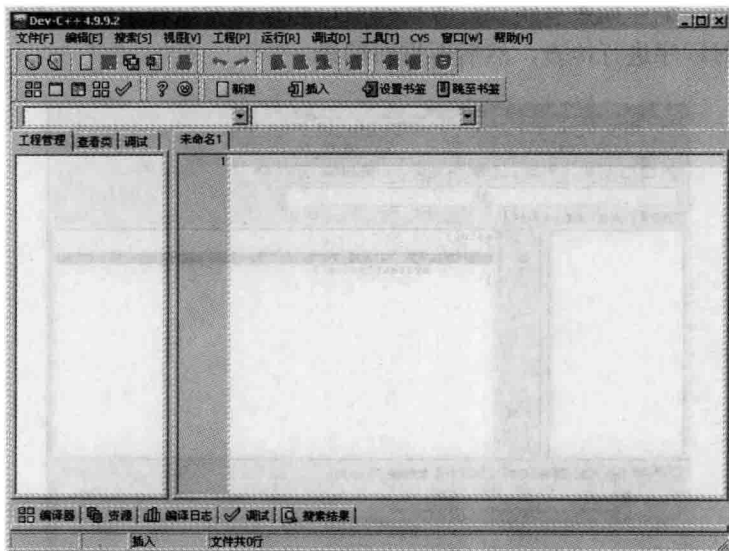


图 1.4 源代码文件编辑窗口

2. 编辑源代码

在代码编辑窗口中输入例 1.1 程序，其编辑方法如同使用记事本一样简单。

3. 保存为 C 程序

选择“文件”|“保存”命令，或使用工具栏中的工具，或按 **Ctrl+S** 组合键，都会打开“保存文件”对话框，如图 1.5 所示。在“保存文件”对话框中输入文件名（如“Ex1”），由于 Dev-C++ 既可以运行 C 程序也可以运行 C++ 程序，所以还要选择文件类型为“C source files”，指定保存的是 C 程序文件。单击“保存”按钮，源代码窗口的名字已经变为“Ex1.c”了。



图 1.5 “保存文件”对话框

4. 编译 C 语言源程序

C 语言程序在运行之前要经过编译和连接，在 Dev-C++ 中是通过编译功能来实现的。选择“运行”|“编译”命令，或使用工具栏中的工具，或按 **Ctrl+F9** 组合键，都可以实现对源程序文件的编译。编译如果有错误，源代码窗口和编译窗口会出现图 1.6 类似的效果，可以根据提示信息对程序进行检查，然后再重新编译。

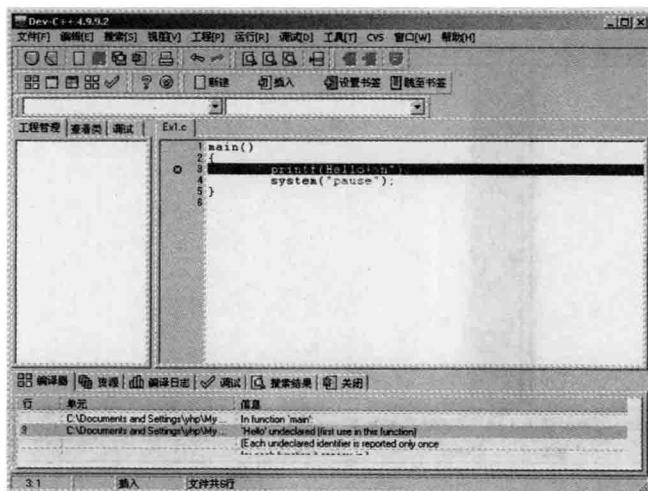


图 1.6 编译出现错误时的状态