中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

高级语言程序设计

——从C到C++

主　编　王兆晖

副主编　周　辉　李红蕾　王泽群

西安电子科技大学出版社

http://www.xduph.com

中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

高级语言程序设计

——从 C 到 C++

主　编　王兆晖

副主编　周辉　李红蕾　王泽群

西安电子科技大学出版社

内 容 简 介

本书涵盖从面向过程编程到面向对象编程的基本内容，通过对 C 语言基本概念、基本语法以及三种基本结构的介绍，读者可领会结构化程序设计的基本原理，并可使用 C 语言进行简单的程序设计；在了解与掌握 C 语言的基础上，通过对 C++的抽象、封装、继承、多态四个特征的介绍，读者可了解面向对象程序设计方法的基本思想。

本书内容浅显易懂，通过对示例程序的深入、明晰的剖析，可使从未接触过程序设计的读者对其有一个基本了解，为以后进一步的专业学习打下良好的基础。

本书可作为学习高级语言程序设计的入门参考书，也可作为高等学校“高级语言程序设计”课程的教材和参考书籍。

图书在版编目(CIP)数据

高级语言程序设计：从 C 到 C++/王兆晖主编. —西安：西安电子科技大学出版社，2016.8
普通高等教育电子信息类“十三五”课改规划教材
ISBN 978-7-5606-4217-8

Ⅰ. ① 高… Ⅱ. ① 王… 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(2016)第 176415 号

策　　划　毛红兵
责任编辑　买永莲
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
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西利达印务有限责任公司
版　　次　2016 年 8 月第 1 版　　2016 年 8 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印　张　17
字　　数　402 千字
印　　数　1～3000 册
定　　价　39.00 元
ISBN 978-7-5606-4217-8/TP

XDUP 4509001-1

前　言

高级语言程序设计是理工科大学生的基础必修课程，国内各高校都普遍开设了计算机语言课，从早期高校开设的 Basic、Fortran、Pascal、C、C++ 到现在的 JAVA、C# 等等，不一而足，而所选择的程序语言多是基于相关的后续专业课程，以期打下一个良好的基础。

自 2006 年起，国内高校陆续开始进行大类招生，如今 100 多所“211 工程”院校中已有多半实行了大类招生。大类招生是高校根据我国教育发展的实际情况做出的教学改革，并不是相近专业的简单归并，而是涉及人才培养模式、课程体系、教学方式方法的一次深刻改革，是学校教学改革的深化和发展，也是学校进行内涵建设，提高人才培养质量的重要举措。

电子信息大类基本上涵盖了原信息类院校的大部分专业，包括电子信息工程、通信工程、计算机科学与技术、信息安全、网络工程等相关专业。这些专业原来都开设有不同的高级语言程序设计课程，在实际教学中，不同的专业，需求各有不同，有的专业偏重 C 语言，比如电子、通信等专业；有的专业则强调面向对象程序设计，比如计算机、软件工程等专业。如何编写适当的教材，使其不仅能够照顾到不同专业的共性需求，切实做到“厚基础”原则，侧重通识教育，而且允许任课教师根据实际情况对内容做灵活的调整，这是非常大的挑战。

C 语言是一种通用的程序设计语言，经过半个世纪的发展，它依然以其精炼的语法、丰富的数据类型在程序设计工具中占有一席之地，而且当前许多程序设计均是用 C 语言实现的。因此，掌握 C 语言不仅可为其他程序设计语言的学习打下良好基础，而且可以起到事半功倍的效果。C 语言不仅能方便地对硬件进行编程，比如驱动程序等，在嵌入式领域应用广泛，也能编写高效的应用程序。C 语言是基础，可以帮助学习者掌握程序设计思想，而 C++ 和 C 语言一脉相承，对初学者来说，相对较为复杂，需要付出大量的时间进行学习和实践，而先从 C 语言开始学习则可以促进对 C++ 的掌握。

作为电子信息大类的计算机公共基础课程，C 语言的课时非常有限，这也是为什么在国内已有大量的关于该课程的书籍的情况下，我们还是从实际需要出发，从学生的基础出发，从该课程的课时出发，并总结了课堂教学经验与学生的反馈，编成了本书。

本书的最大特点是将 C 语言的结构化程序设计作为基础，通过浅显易懂的内容讲解，使学生可以尽快地完成从不会编写程序到可以开始实践的过程，然后，在学生对 C 语言基本掌握的基础上又引入 C++ 面向对象程序设计的基本概念，使学生能够对程序设计的发展、程序设计思想的变迁，以及高级语言程序设计有个基本的了解和掌握。

本书的前八章介绍了 C 语言的基本数据类型、自定义数据类型、三种基本结构以及数组、函数、指针的基本概念，后五章介绍了 C++ 面向对象程序设计的思想及四个基本特点——抽象、封装、继承、多态。

本书在编写过程中得到了海南大学信息学院高级语言程序设计教学团队的大力支持，王兆晖负责对全书进行统筹规划，并完成了第 1～7 章及第 9 章的编写；王泽群对前九章进

行了完善补充，并完成了第 8 章的编写；李红蕾负责第 10、11 两章的编写；周辉负责第 12、13 两章的编写。

本书在编写的过程中，得到了孙盛杰、彭金莲、周星、任一凡、林志阳、吴泽晖、胡祝华、卢春燕、邢怡杏等老师在技术上的鼎力相助，在此向他们致以衷心的感谢。本书的编写也得到海南省科技合作专项(KJHZ2015-23)的大力支持，在此表示衷心的感谢。

由于编者水平所限，书中疏漏与不妥在所难免，敬请广大读者批评指正。

编　者

2016 年 6 月

海口

目　录

第 1 章　C 语言入门

万事开头难，对于初次接触 C 语言的读者，本章我们将通过对相关内容的简单介绍，使读者对 C 语言有一个基本概念，对使用 C 语言进行程序设计的流程有一个初步的了解。因此在内容安排上，不会纠结于细节描述，而是通过对一些简单程序的编写及执行，使读者体会到本书各章节内容的不同作用，以及这些内容是如何集成到一个整体中，从而形成一个完整的 C 语言程序并实现特定功能的。

1.1　初识 C 程序

任何自然语言的学习基本上都是由简入繁，通过对一些基本词汇、语法的学习而逐步掌握并应用的。C 语言的学习也是如此。对于初学者来讲，C 语言的词汇即关键字，接近于我们使用的自然语言(这里指英语)，在某种程度上可以顾名思义。下面就让我们通过最简单的一个程序来认识和了解 C 语言。

【例 1.1】 在屏幕上输出一行文字：欢迎使用 C 语言！

具体程序：

```
//例题 1.1，第一个 C 语言程序
#include <stdio.h>
//主函数 main( )
int main( )
{
    printf("欢迎使用 C 语言！ ");      //屏幕上显示输出一段文字
    return 0;                          //程序成功执行
}
//主函数结束
```

对于上述代码，我们需要进行编辑、保存、编译和运行，从而在屏幕上输出相应的文字。编辑代码，必须有一个可以进行文字编辑的软件。Windows 操作系统中自带的记事本或者 MS Word 等软件，均可以完成代码的编辑与存储功能。在 C 语言程序设计中，程序代码编写完成后保存在扩展名为“.c”的文件中，该文件即源程序文件。我们将上述代码保存在 Lianxi_1_1.c 源程序文件中。

因为 C 语言是高级语言，因此需要将源程序翻译成计算机可以识别的机器语言，也就是要对源程序进行编译。程序在成功编译以后，即可以运行，在屏幕上输出相应的文字，如图 1.1 所示。

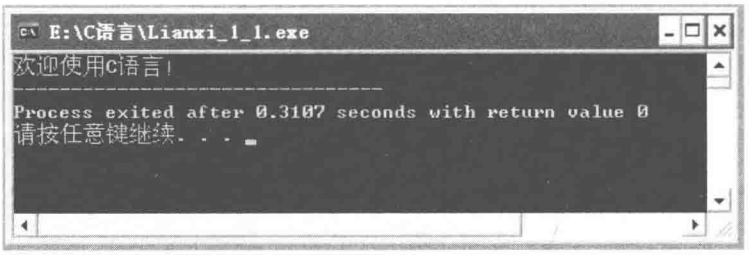


图 1.1　例 1.1 程序执行结果

程序执行后，输出结果在 Windows 的 Console 控制台(也称作 Windows 的命令行 DOS 框)中，本书中简称为控制台。

运行结果中，在标题栏列出了当前运行的程序“Lianxi_1_1.exe”，该程序即源程序编译之后的可执行程序。控制台上显示出要求输出的文字“欢迎使用 C 语言!”，以及该程序执行所用时间和返回值。当根据控制台上的提示“请按任意键继续...”操作时，该程序运行结束。

虽然例 1.1 仅有短短数行程序代码，但是已经具备了 C 语言的基本框架和特征。源程序中符号“//”后面的内容为注释，注释的作用是在源程序中对特定的代码进行简要解释，起到标记相应代码的作用，以增强程序的可读性，尤其是在复杂的程序源文件中，注释可起到相当于文档的辅助作用。当程序运行时，注释不会对程序产生任何作用。

C 语言中的注释有两种形式：

(1) 单行注释：使用双斜线符号“//”注释其后的内容，表示双斜线以后到本行结束的内容均为注释部分。

(2) 多行注释：使用一对符号“/*”与“*/”完成内容注释，表示从符号“/*”开始到符号“*/”结束之间的内容均为注释，无论这些内容是在一行还是多行中。

单行注释短小精悍，多用于代码中对某语句的解释。多行注释可以详细描述某个部分的具体功能。C 语言编译器在编译源程序时，对注释部分忽略不计，不会为注释生成机器代码。因此详尽的注释，是学习 C 语言程序设计时需要养成的一个良好的编程习惯，在任何时候，详尽的注释都可以帮助程序设计人员迅速了解该代码的含义或者功能。

源程序的第 2 行以“#”开始的部分，是一个预处理指令。该指令在对程序进行编译之前告诉预处理器该程序要包括标准输入/输出函数库。其中“stdio”是“standard input&output”的缩写，扩展名“.h”中的“h”是英文单词 head 的缩写，即头文件。C 语言头文件 stdio.h 中包含与输入/输出相关的函数，在本程序中使用的 printf( )函数即定义在该文件中。因此凡是涉及与输入/输出相关的函数时，需要确认标准函数库中是否已经定义，若已定义，则可以直接使用，并在源程序中应用上述预处理指令。关于预处理指令，本书第 11 章将会进行详细介绍。

代码“int main( )”是任何 C 语言程序必不可少的一部分。C 语言程序由一系列的函数所构成，C 语言程序设计的基本单位是函数，这些函数可以在一个源文件中，也可以根据功能不同分布在多个源文件中。在这一系列的函数中必须有一个而且只能有一个以 main 为名的函数，这个函数称为主函数，整个程序由主函数开始执行，一般而言，也是在主函数结束。该行代码中，关键字 int 表示该主函数需返回一个整型数值，与程序最后一行的“return

0;”相对应。

代码“printf("欢迎使用 C 语言！");”告诉计算机执行一个输出动作，是一条可执行语句。“printf”为“print function”的缩写，是一个格式化输出函数，该函数在标准输入输出文件 stdio.h 中定义。该函数将双引号中的内容输出到屏幕上。C 语言的语法规定分号“;”是一条语句的终止符号，每一条可执行语句都必须以分号结束，但是只要符合语法，多条语句可以在一行书写，一条语句也可以在多行书写。

1.2 C 程序从编辑到运行

我们从上一节的介绍中可以看到，即使一个简单的 C 语言程序，从编写到运行也经过了几个必不可少的过程，如图 1.2 所示。

图 1.2 C 语言程序从编写到运行的过程

编辑阶段，使用文字处理软件设计编写代码，并将文件保存为以“.c”为扩展名的源程序文件。编译阶段，编译器(Compiler)将用 C 语言编写的源程序翻译为计算机可识别和执行的机器指令，即目标程序(以“.obj”为扩展名)。和源程序一样，目标程序也不能直接执行，还需要经过连接器(Linker)，同 C 语言库中提供的支持程序和其他相关目标模块连接起来，然后就生成了可以直接运行的可执行文件(以“.exe”为扩展名)。我们在上一节中提到的标准输出函数 printf( )，即在连接阶段通过对标准函数库的连接，与源程序结合成了可执行文件，所以连接器为 C 语言提供了丰富的手段，通过与外在资源的连接，迅速扩充了 C 语言，从而得到了功能强大的可执行程序。

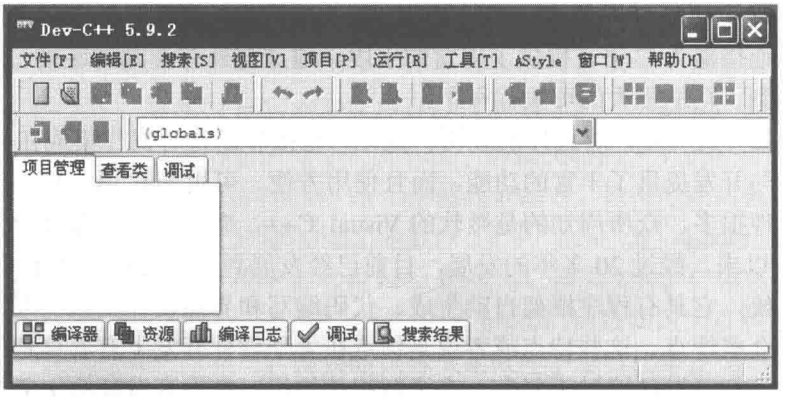
C 语言集成开发环境，将程序的编辑、编译、连接和运行等操作集成在一起，为 C 语言的程序设计与开发提供了丰富的功能，而且使用方便。可用于 C 语言程序开发的商业软件或者开源软件很多，众所周知的是微软的 Visual C++。微软公司自 20 世纪 90 年代初推出 Visual C++ 以来，经过 20 多年的发展，目前已经发展到 Visual C++ 2013 版。作为可视化集成开发系统，它具有程序框架自动生成、代码编写和界面设计集成交互、高级除错、可远程调试等众多特点，这些特点或者说是优点在大型软件开发上有着卓越的表现，但对于初学者而言，其开发环境过于复杂，在了解相关知识、掌握开发环境上需要投入的精力与时间，将远远超过学习 C 语言所花费的时间，且作为商业软件，Visual C++ 价格较高，因此对于初学者，我们不推荐使用该开发环境。

在网络上还存在着许多开源软件，包括历史久远的 Turbo C、MinGW 等免费集成开发环境。本书推荐初学者使用 Dev-C++。Dev-C++ 是一个 Windows 环境下的集成开发环境，可用于 C 语言或者 C++ 语言程序的开发。作为一款自由开源软件，它集合了众多 C/C++ 语言开发环境的优点，且免费、实用。Dev-C++ 遵循 C11 标准并兼容 C99 标准。其开发环境

包括多页面窗口、工程编辑器以及调试器等，在工程编辑器中集成了编辑器、编译器、连接程序和执行程序，提供高亮度语法显示以减少编辑错误，具备基本的调试功能，可以进行单步语句跟踪调试，非常适合初学者。对于初学者而言，基本不需要花费太多的时间，就可以投入到C语言的程序设计练习中，而不用纠结于各种开发环境参数设置或背景知识的学习。

最重要的是Dev-C++有中文界面，这也是目前很多C语言开源软件不具备的优势。该软件可通过网络免费下载使用。在下载了该软件之后，双击该软件，软件即开始自动安装。在安装完成后第一次运行时，需要对运行环境进行配置，即选取开发界面所使用的语言，如图1.3(a)所示。这时如果选择中文，软件的开发界面则为中文模式(图1.3(b))。

(a) 界面语言的选择



(b) 初始开发界面(中文模式)

图1.3 Dev-C++运行环境的配置

下面我们通过例1.1来介绍如何使用Dev-C++，实现从源程序编辑到程序执行的完整过程。

1. 源程序编辑

如图1.4所示，选取【文件】→【新建】→【源代码】菜单项，进入源代码的编辑界

面。按照例 1.1 的内容输入代码，并保存为 Lianxi_1_1.c 源程序文件。

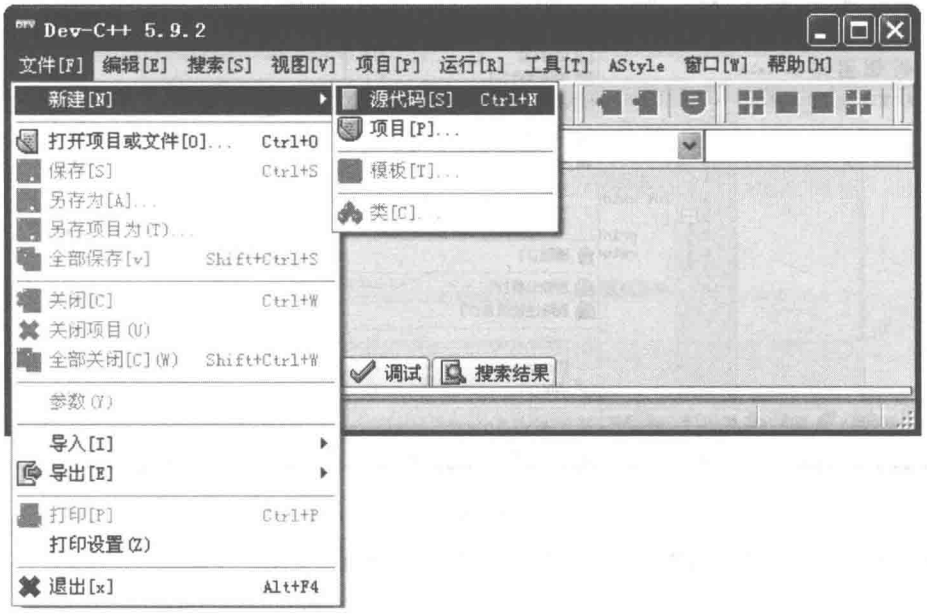


图 1.4　新建一个源程序编辑页面

在学习 C 语言程序之初，应该形成良好的编程习惯，除了前文提到的对代码进行详尽的注释之外，还应该将代码编写得错落有致，不同部分之间的行距间隔如图 1.5 中所示，这样程序的可读性也会随之提高。

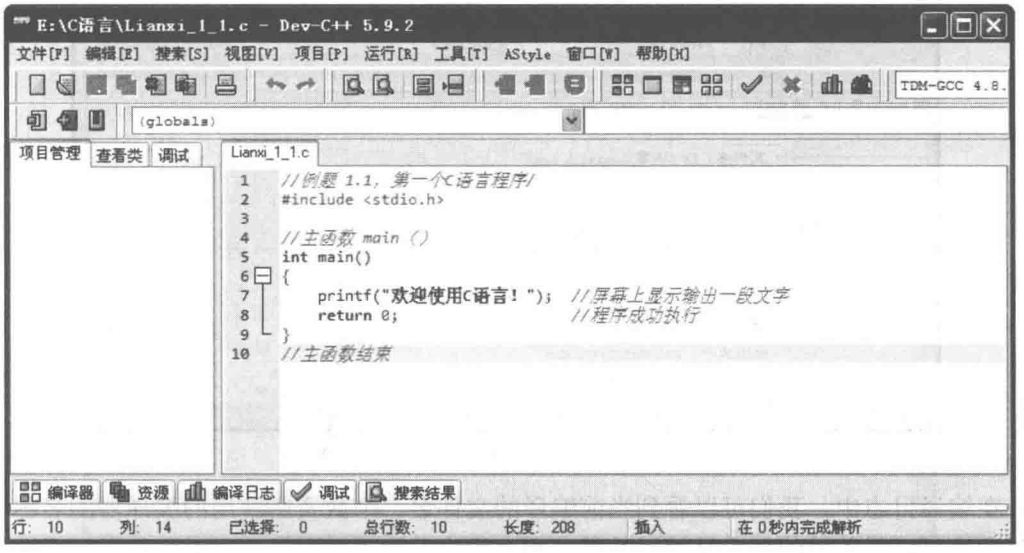


图 1.5　编辑源程序

2. 源程序编译

在程序编写完成后，选取【运行】→【编译】菜单项，如图 1.6 所示，即开始对源程序进行编译。

图 1.6　编译源程序

编译过程中相关信息的反馈以及编译结果会显示在开发环境的编译日志中，如图 1.7 所示。

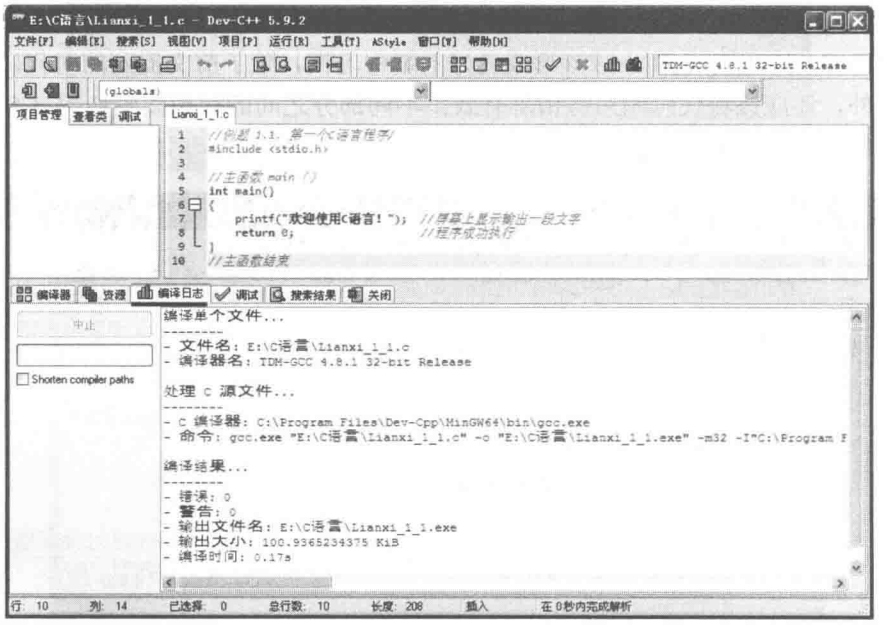


图 1.7　编译日志

在编译日志中，我们可以看到当前编译的文件名、个数及所使用的编译器的名称。对初学者来讲，最关键的部分是编译结果。因为程序的设计编写，不可避免地会出现这样或那样的错误。编译过程中会进行 C 语言的语法检查，对于没有遵守语法规则所引起的错误，编译器会显示出错误信息，从而使程序编写人员可以很容易地定位该错误并进行修正。在图 1.7 中，编译器对 Lianxi_1_1.c 文件成功地进行了编译，编译结果中没有错误信息，并成功生成了可执行文件 Lianxi_1_1.exe。

如果我们将例 1.1 中 printf 语句后的分号去掉，再进行编译，就会看到，如图 1.8 所示，在开发界面下部的编译器窗口中，显示了错误信息，并且指明了错误产生的位置，在函数 main 中，第 8 行的“return”之前，应该有分号“;”。

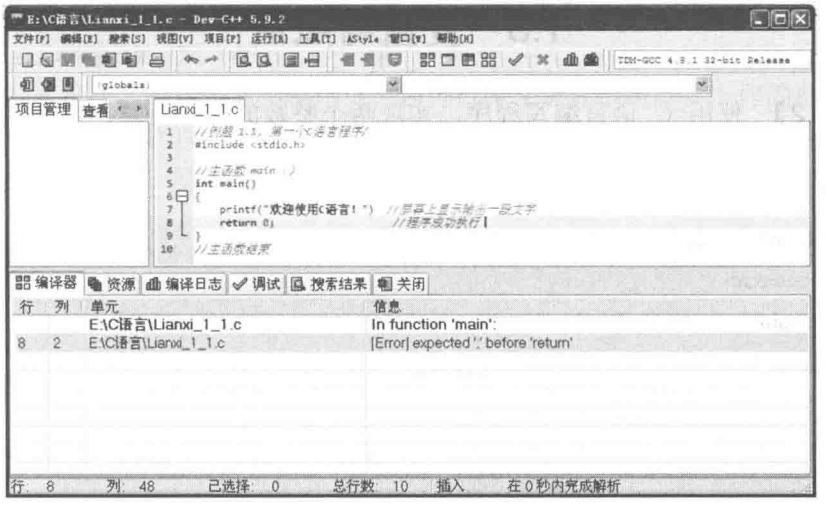


图 1.8　编译错误示例

同时，在编译日志中，也相应地指出了程序错误及错误发生的位置，如图 1.9 所示。在编译结果中，显示出错误发生的数目(1 个)，由于编译过程中有错误产生，编译没有生成可执行程序。

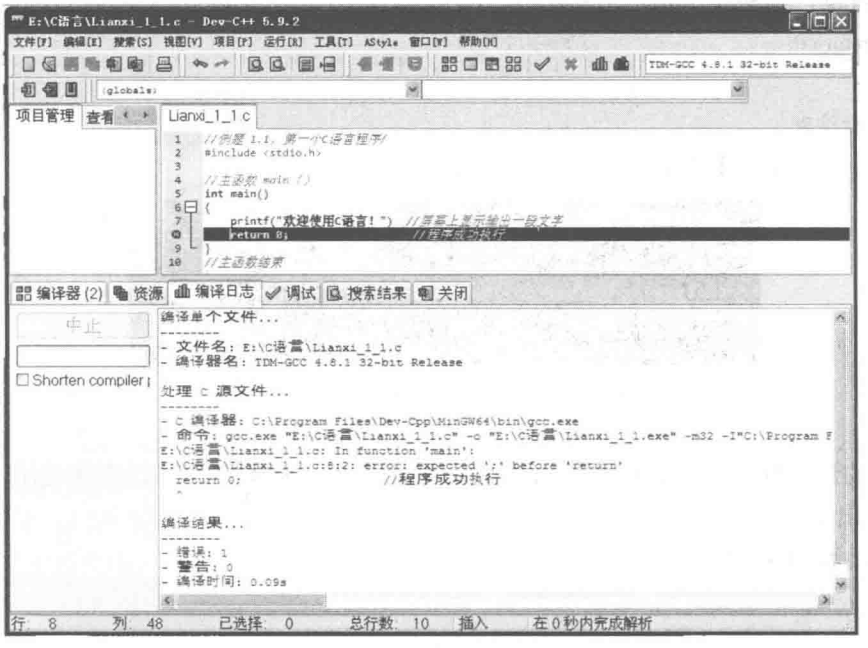


图 1.9　编译日志中反映的编译错误信息

3. 程序执行

当源程序编译成功之后，即生成了相应的可执行文件，选取【运行】→【运行】菜单

项，可运行该程序。此时，运行的是编译生成的可执行文件，而不是源程序文件。这时Windows 控制台窗口弹出，显示出运行结果，如图 1.1 所示。

1.3 变量与赋值

【例 1.2】 使用 C 语言编写程序，实现两个整数的加法运算，并将结果显示在屏幕上。

具体程序：

```
//例题 1.2，两个整数求和
#include <stdio.h>
//主函数 main( )
int main( )
{
    int a;                  //第一个整数
    int b;                  //第二个整数
    int sum;                //和
    a=23;
    b=24;
    sum=a+b;
    printf("sum=%d",sum);   //将求和的结果输出
    return 0;               //程序成功执行
}
//主函数结束
```

编译完后，运行程序，输出结果，如图 1.10 所示。

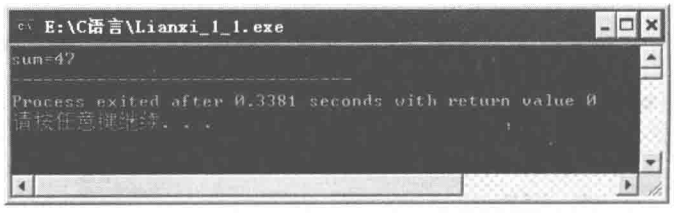


图 1.10 两个整数的求和结果

程序的预处理指令引入标准输入/输出函数库，因此在程序中可以直接使用 printf( )标准输入/输出函数。

在主函数 main( )中，语句

```
int a;
int b;
int sum;
```

定义了三个变量，名称分别是 a、b、sum。

所谓变量，就是程序中用来存放数据的对象，在程序运行期间其值可以改变。每一个

变量都应该有一个名字，即变量名，便于在程序设计中使用。C 语言规定了所有变量在使用前必须先定义，包括定义变量的数据类型和变量名。通常，变量的定义会放在函数的开始部分、其他可执行语句之前。在例 1.2 中，我们定义了三个变量 a、b、sum，这些变量的数据类型是 int，表示这三个变量只能存放整型数据(即整数)。在程序执行过程中，这三个变量分别存放了整数 23、24、47。

C 语言的变量除了可以定义为整型外，还可以定义为其他类型。不同的数据类型，对应的存储空间不同，存储的数据也不同。在下一章将对数据类型进行详细介绍。

在定义变量的时候，需要遵从 C 语言的语法规则：变量的定义至少应包含变量的数据类型与变量名，即

```
数据类型    变量名
```

也可以通过逗号间隔，定义同一类型的多个变量：

```
数据类型    变量名 1，变量名 2，…，变量名 n
```

根据上述规则，例 1.2 中的三个变量可以分别定义如下：

```
int a;
int b;
int sum;
```

由于这三个变量均为整型变量，也可以定义为

```
int a, b, sum;
```

变量必须先定义后使用，如果变量在使用之前没有定义，则在对程序进行编译时，编译器会返回错误信息，指出该变量在使用前没有定义。C 语言强制要求变量在使用之前被定义，可以起到以下作用：

(1) 在定义变量时明确指定了数据类型，程序在编译时，就能为该变量分配相应的存储空间，并检查该变量的使用是否正确。

(2) 在定义变量时，除了指定数据类型外，还需要为变量命名，如例 1.2 中的变量名 a、b 和 sum。为变量命名就好像为饭店的每一个包厢指定名称一样，该名称要便于程序设计人员在程序中使用。C 语言规定了变量的命名需要遵循一定的规则：

① 变量名只能由字母、数字、下划线组成。

② 变量名的第一个字符只能是下划线或者字母。

根据上述两项规则，我们可以判断下列变量名是否为合法的变量名：

Lower，3G，_command，firstName，FirstName，%c

其中，变量名“3G”不符合第二项规则，而变量名“%c”既不符合第一项规则也不符合第二项规则，因此这两个变量名是非法的。

此外，C 语言对字母的大小写敏感，即认为大写字母与小写字母是两个不同的字符。因此在上述变量名中，firstName 与 FirstName 是两个不同的变量名。除了上述两项规则以外，C 语言的相关标准并没有规定变量名的长度，以及如何给变量命名，所以变量名的选择相对来说是自由的。例 1.2 中，第一个整数变量，既可以命名为 a，也可以命名为 aok 等。而且 C 语言中也没有限定变量名的长度，理论上讲变量名可以无限长。但是为了程序书写的便利以及阅读的方便，变量名应控制在有限长度内，并且变量名应该有一定的含意(与英文含意相对应)，这样在阅读程序时，可以很容易地判断出该变量的用途。比如为了存储两