

 普通高等教育计算机规划教材

C++程序设计教程

实验指导及习题解答

第 2 版

刘慧宁 孟威 等编著



机械工业出版社
CHINA MACHINE PRESS



普通高等教育计算机规划教材

C++程序设计教程实验 指导及习题解答

第2版

刘慧宁 孟威 等编著



机械工业出版社

本书是“普通高等教育计算机规划教材”中《C++程序设计教程》的配套上机实验指导与习题解答。本书各章均包括上机实验指导和习题解答两部分。其中，上机实验指导部分配合教材的关键知识点设计了相关实验；习题解答部分给出了配套教材中全部习题的答案。本书附录给出了 3 个综合实验，旨在提高学生的实际编程能力。本书所有程序均在 Dev-C++ 下调试通过。

本书可作为高等院校、计算机水平考试培训、各类成人教育 C++ 程序设计课程的配套教学辅导书，也可供自学的读者参考使用。

图书在版编目 (CIP) 数据

C++程序设计教程实验指导及习题解答 / 刘慧宁, 孟威等编著. —2 版. —北京: 机械工业出版社, 2009.3

(普通高等教育计算机规划教材)

ISBN 978-7-111-26629-7

I. C… II. 刘… III. C 语言—程序设计—高等学校—教学参考资料
IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 042127 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: 张宝珠 罗子超

责任印制: 杨 曦

保定市 中画美凯印刷有限公司印刷

2009 年 5 月第 2 版·第 1 次印刷

184mm×260mm·16 印张·392 千字

0001—3000 册

标准书号: ISBN 978-7-111-26629-7

定价: 26.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294 68993821

购书热线电话 (010) 88379639 88379641 88379643

编辑热线电话 (010) 88379753 88379739

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- (1) 反映计算机技术领域的新发展和新应用。
- (2) 注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。
- (3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- (4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- (5) 适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

前 言

本书是“普通高等教育计算机规划教材”中《C++程序设计教程》（第3版）的配套上机实验指导与习题解答。

计算机程序设计是一门实践性很强的课程。一些初学者往往觉得已经看懂了教材，但在上机实践遇到问题时却束手无策，因此，学习积极性受到打击。其实，只要坚持多编写一些程序，就会找到编程的感觉，对教材中的内容有进一步的理解。

相对本书第1版，本书代码均用现代标准C++风格重新改写，并在Dev-C++下调试通过，同时增加了3个综合实验。本书各章均包括两个部分，第一部分是上机实验，其中的基本实验与教材各章同步，包括1个或者多个实验，每个实验都包括题目、实验目的、实验步骤和参考答案部分；第二部分是教材中全部习题的答案，对于编程题目给出了参考程序。本书附录给出了3个综合实验，目的在于锻炼学生综合运用所学知识解决问题的能力。

本书在强调与教材紧密配合的同时，为使初学者能尽快入门，在上机实验环节中加入Dev-C++上机环境，初学者可以按照书中给出的步骤完成程序的编辑、编译、连接与运行。当然，这也是一般高级语言的上机过程。

尽管本书给出了教材习题的解答和上机实验的参考答案，但希望读者能先试着独立完成教材习题和上机实验题目，然后再参考本书答案。一般来说，一个问题的编程解决方法并不是唯一的，书中答案也不见得是最好的，也许你的方法更有效、更完美。

本书第1~6章由刘慧宁编写，第7~12章由孟威编写。参与本书实验部分编写工作的有那盟、孙连英、孟勇、魏庆阳、刘蕾、刘光华、刘兰兰、刘晓杰、李清华，参与本书习题解答部分代码编写与调试工作的有祁磊、满欣、孙岩松、邵洋、张庆等。

书中如有不当或错误之处，欢迎读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 概述	1
1.1 基本实验	1
1.1.1 编辑、编译、链接和运行 C++ 程序	1
1.1.2 调试 C++ 程序	7
1.2 习题解答	10
第 2 章 C++ 语言编程基础	13
2.1 基本实验	13
2.1.1 运算符与表达式	13
2.1.2 选择语句	14
2.1.3 循环语句	15
2.1.4 数组操作	16
2.2 习题解答	17
第 3 章 函数	31
3.1 基本实验	31
3.1.1 函数定义	31
3.1.2 函数递归	32
3.1.3 函数重载	33
3.2 习题解答	34
第 4 章 类与对象	48
4.1 基本实验	48
4.1.1 类和对象的定义	48
4.1.2 类的组合	52
4.2 习题解答	54
第 5 章 指针与引用	74
5.1 基本实验	74
5.1.1 传址与引用调用	74
5.1.2 函数类型的参数	76
5.2 习题解答	78
第 6 章 继承	107
6.1 基本实验	107
6.1.1 单继承	107
6.1.2 多继承	108
6.2 习题解答	112

第7章 运算符重载	123
7.1 基本实验	123
7.2 习题解答	126
第8章 多态与虚函数	139
8.1 基本实验	139
8.2 习题解答	143
第9章 容器类	155
9.1 基本实验	155
9.2 习题解答	159
第10章 模板与标准模板库	173
10.1 基本实验	173
10.2 习题解答	175
第11章 输入/输出流	188
11.1 基本实验	188
11.2 习题解答	190
第12章 异常处理	205
12.1 基本实验	205
12.2 习题解答	206
附录 综合实验	214

第1章 概 述

1.1 基本实验

1.1.1 编辑、编译、链接和运行 C++程序

实验题目

使用 Dev-C++ 4.9.9.2 集成开发环境编辑、编译、链接和运行 C++程序。

实验目的

- (1) 熟悉 Dev-C++ 4.9.9.2 集成开发环境。
- (2) 掌握使用 Dev-C++ 4.9.9.2 对简单 C++程序进行编辑、编译、链接和运行。

实验步骤

- (1) 启动 Dev-C++ 4.9.9.2 集成环境。
- (2) 建立并保存源文件。
- (3) 编译并链接该源程序文件，生成可执行文件。
- (4) 运行可执行文件。

参考答案

Dev-C++是一个可视化集成开发环境，可以使用此软件实现 C/C++程序的编辑、编译、链接和运行。

1. 启动 Dev-C++

启动 Dev-C++，最基本的方法是：

- (1) 单击任务栏中的“开始”按钮，选取“所有程序”（或“程序”）菜单项，然后选取子菜单项“Bloodshed Dev-C++”，显示该项下的子菜单。
- (2) 单击“Dev-C++”菜单项，即可启动 Dev-C++集成开发工具，如图 1-1 所示。

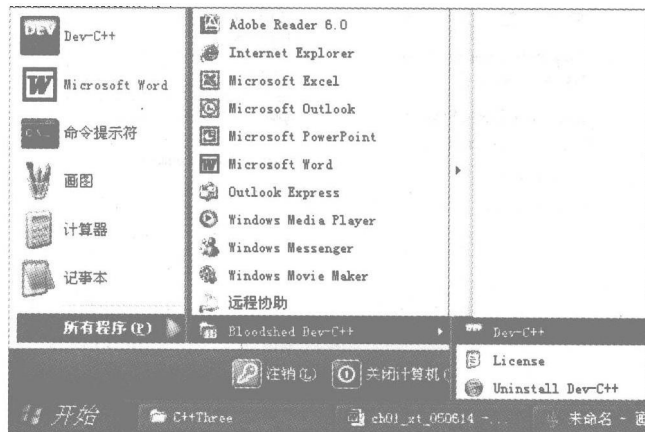


图 1-1 启动 Dev-C++

如果用户桌面上建立了相应的快捷方式，启动 Dev-C++则更简单，方法是：直接双击桌面上的 Dev-C++图标即可。

启动 Dev-C++后的初始界面如图 1-2 所示。

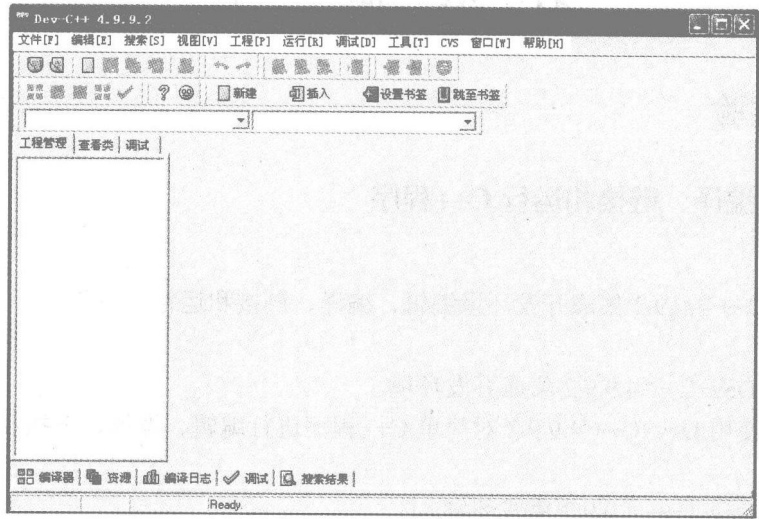


图 1-2 启动 Dev-C++后的初始界面

如果读者看到的是英文界面，可以将它改为中文。具体操作是：依次单击主菜单“Tools”→“Environment Options”菜单项，在弹出的对话框中选择“Interface”选项卡；然后，在“Language”下拉列表中选择 Chinese，如图 1-3 所示；最后，单击“OK”按钮，界面就变成中文界面。其他语言的界面也可通过相似的方法改为中文界面。

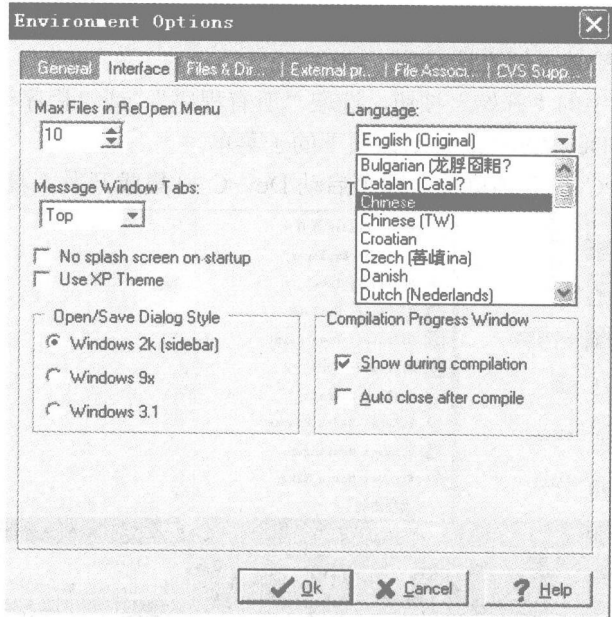


图 1-3 将英文界面改为中文

2. 建立与保存源文件

启动 Dev-C++ 后，就可以使用它建立源文件。要建立一个新的源代码文件，应该首先单击主菜单“文件”，选取“新建”菜单项，然后单击“源代码”子菜单项，如图 1-4 所示。

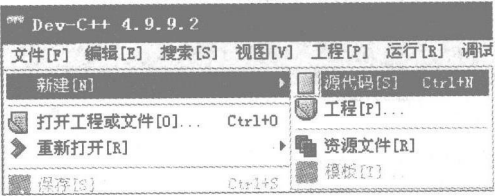


图 1-4 准备新建源文件

单击“源代码”子菜单项后，屏幕右下侧出现一片白色区域，可在此处输入程序源代码。如图 1-5 所示。图 1-5 是假设读者已经输入了习题 5 参考答案中给出的程序源代码。

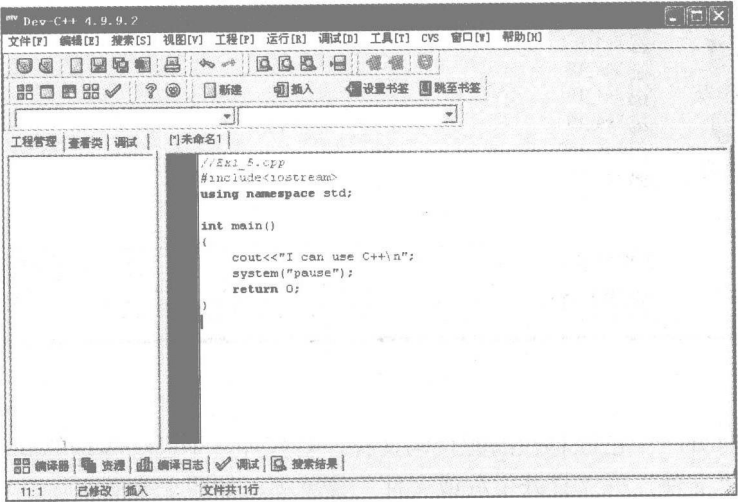


图 1-5 新建源文件

编辑源代码时，一个好的习惯是创建了一个新的程序源代码文件后，在输入部分代码之后先将该文件保存到硬盘某个目录下，然后在文件的编辑过程中经常性地保存，以防止机器突然断电或者死机。要保存文件，只需依次单击主菜单“文件”→“保存”菜单项，如图 1-6 所示。

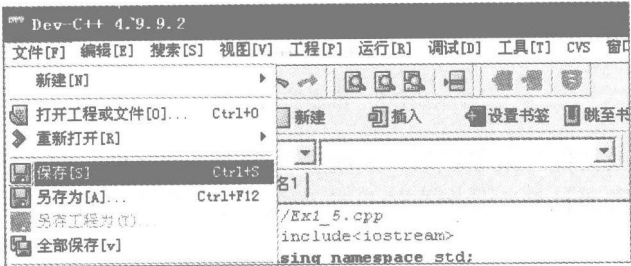


图 1-6 准备保存源文件

单击“保存”菜单项后，系统会弹出“保存文件”对话框，如图 1-7 所示。在此对话框中，需要指定文件将要存放的目录（图示为 book 文件夹）、文件名称（图示为 Ex1_5）及保存类型（对于 C++ 程序，应为 C++ source files）。单击对话框中的“保存”按钮后，在 book 目录下将会出现一个名为 Ex1_5.cpp 的源文件。

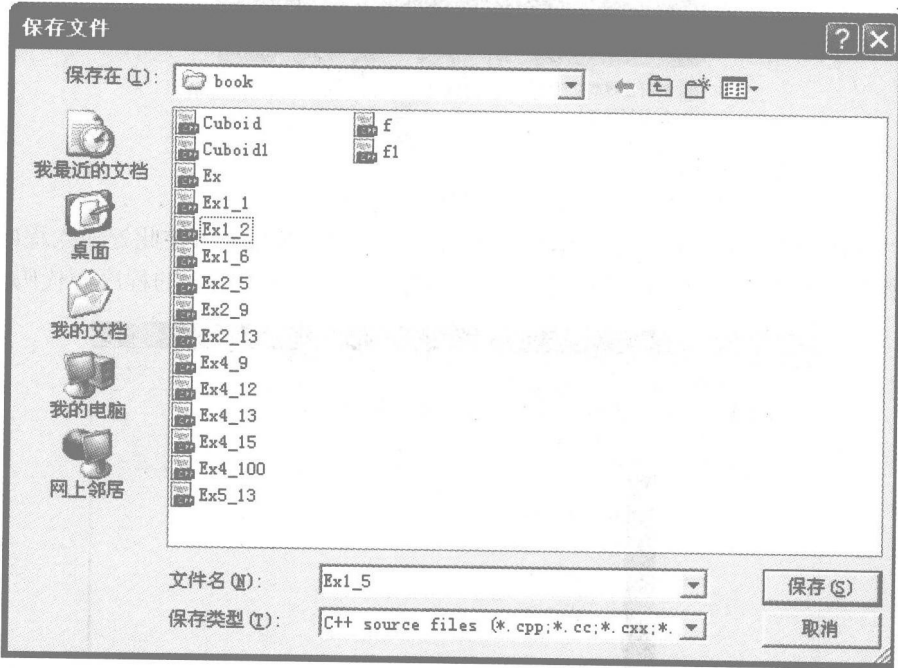


图 1-7 “保存文件”对话框

读者也可以打开一个已经存在的源代码文件。要打开一个已经存在的源代码文件，只需依次单击主菜单“文件”→“打开工程或文件”菜单项，如图 1-8 所示。

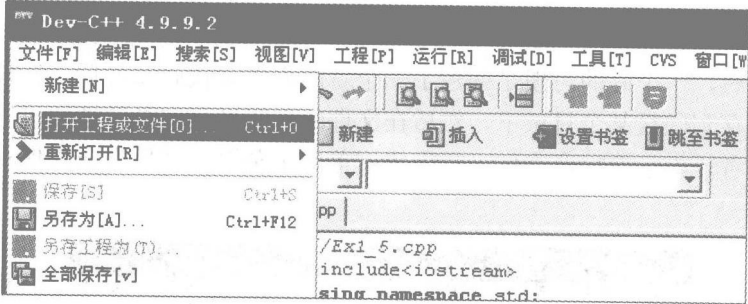


图 1-8 准备打开文件

单击“打开工程或文件”菜单项后，系统会弹出“打开文件”对话框，如图 1-9 所示。在此对话框中，指定需要打开文件所在的目录及文件名，单击“打开”按钮，该文件就会被打开。

对于新建或已经存在的源程序文件都可以进行编辑，Dev-C++提供了通常的源文件编辑功能。当拖动鼠标选择一部分文件内容后，就可以激活主菜单“编辑”中的常用编辑选项。

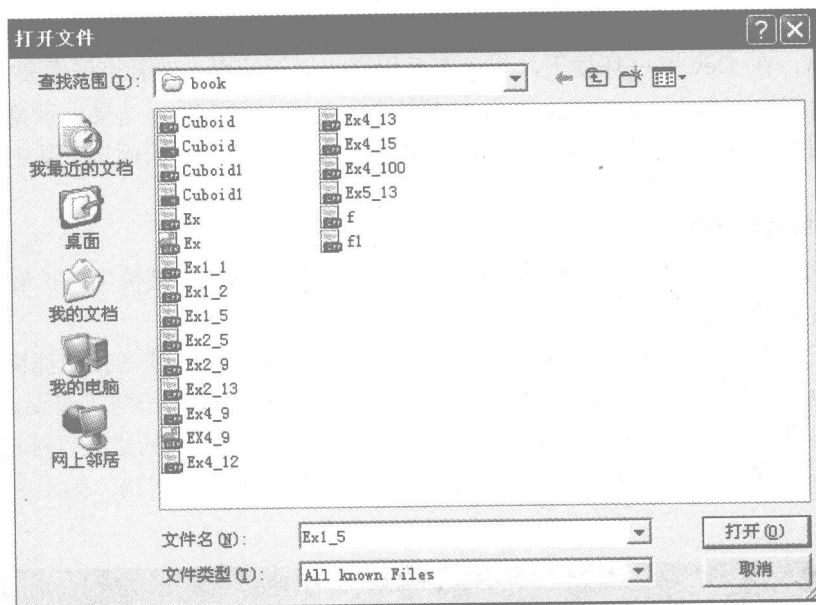


图 1-9 “打开文件”对话框

如图 1-10 所示，选择了一行代码，被选中的代码反相显示。此时，可以通过主菜单“编辑”中的编辑选项或者通过右击鼠标弹出的菜单进行各种编辑操作，如剪切、复制、粘贴、删除等。这些操作与各种常见文本编辑软件的方法类似，如图 1-10 所示。

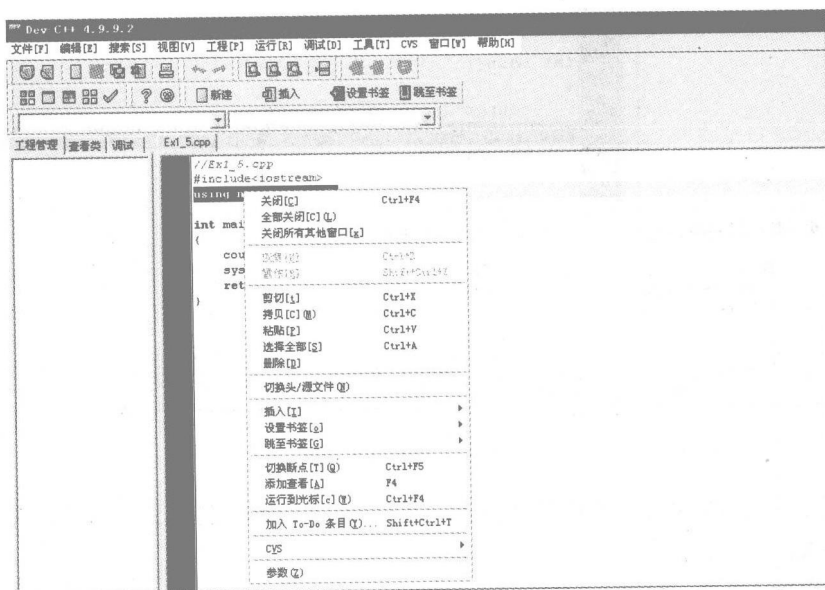


图 1-10 “编辑”菜单选项

在程序编辑过程中，应该随时进行保存，保存方式与前面所述相同。不过，除第 1 次保存文件外，保存时不会弹出对话框，此时，程序文件将以同一文件名重新保存到之前指定的目录下。如果想将程序保存到其他的硬盘目录下或保存为其他文件名，可以依次单击主菜单

“文件”→“另存为”菜单项，在弹出的对话框中，重新指定程序文件的名称和保存目录。

需要指出，在 Dev-C++环境下，为了查看程序的运行结果，读者可能需要在 main 函数的 return 语句前加上“system("pause");”，这样程序运行到该语句时，结果显示窗口将会停留，让人们有时间查看程序的输出结果；否则，结果显示窗口可能会一闪而过。其中，pause 不区分大小写。

3. 编译与链接程序

C++程序的源代码文件是不能直接运行的，它必须经过编译、链接生成扩展名为.EXE 的可执行文件后才能够运行。

在 Dev-C++环境下，依次单击主菜单“运行”→“编译”菜单项（或者按快捷键 <Ctrl+F9>），可以一次性地完成程序的编译和链接过程。如果程序中存在词法、语法等错误，则编译过程失败，编译器将会在屏幕下方显示错误信息，并且将源程序中相应的错误行标成红色底色，如图 1-11 所示（程序中由于删除了 cout 语句后面的分号，编译时报错，提示 system 语句前面缺少分号）。

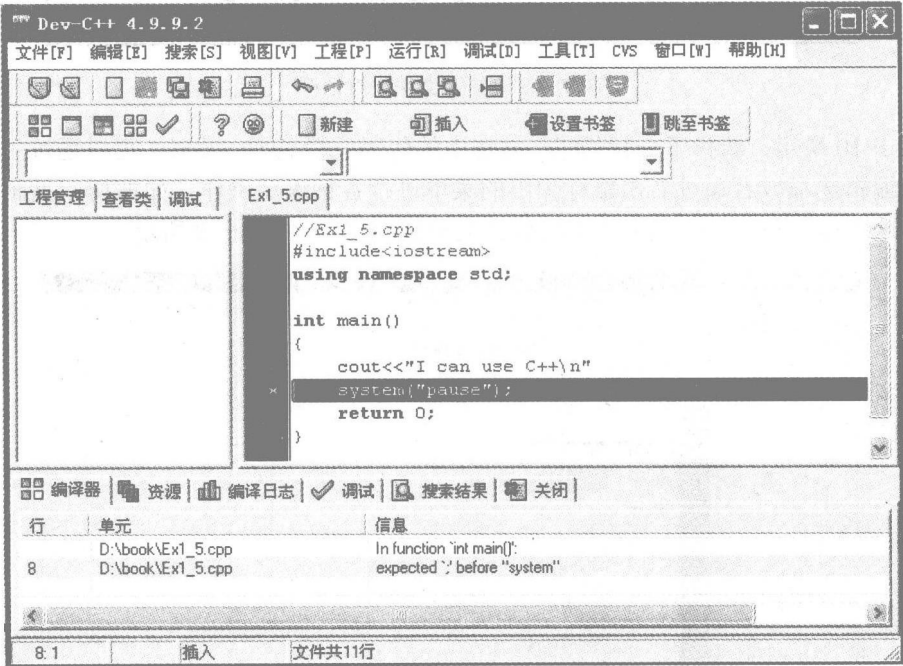


图 1-11 编译错误信息

改正程序中存在的词法、语法等错误后，编译成功。此时在源文件所在目录下将会出现一个同名的.EXE 可执行文件。

4. 运行程序

程序编译、链接成功后，生成的可执行文件可以通过多种方法运行。通常采用的方法有 3 种：

- (1) 双击生成的.EXE 文件。
- (2) 在 DOS 窗口直接运行。

(3) 在 Dev-C++环境下, 依次单击主菜单“运行”→“运行”菜单项(或者按快捷键<Ctrl+F10>)运行, 如图 1-12 所示。

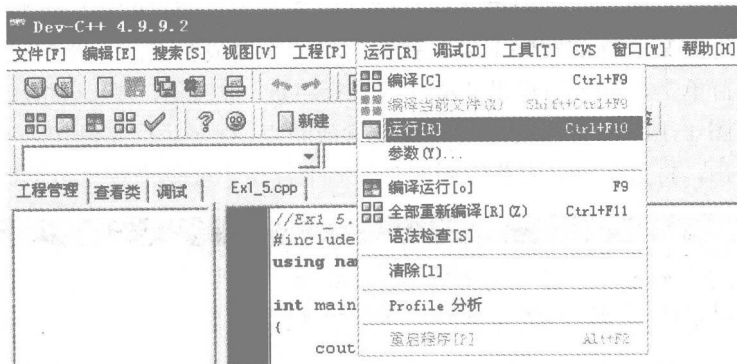


图 1-12 运行程序

程序运行后的输出结果如图 1-13 所示。

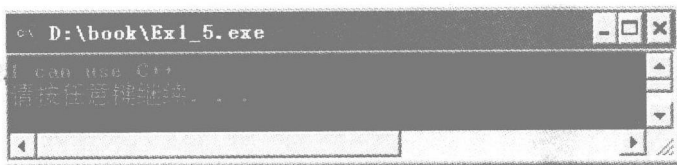


图 1-13 结果显示窗口

按任意键, 即可退出程序的运行, 关闭结果显示窗口。

1.1.2 调试 C++程序

实验题目

使用 Dev-C++ 4.9.9.2 集成开发环境调试 C++程序。

实验目的

- (1) 熟悉 Dev-C++ 4.9.9.2 集成开发环境。
- (2) 掌握使用 Dev-C++ 4.9.9.2 对简单 C++程序进行调试。

实验步骤

- (1) 建立并保存源文件。
- (2) 编译并链接该源程序文件, 生成可执行文件。
- (3) 调试该程序。

参考答案

程序源代码通过编译和链接, 只是表明该程序中没有词法和语法错误, 运行时还是有可能出错(比如, 算法不对导致运行结果不正确)。程序运行出错时, 同样需要找出错误原因。对于比较简单的程序, 仔细阅读源代码一般能找出其中的错误, 但对于复杂程序, 仅靠阅读源代码往往解决不了问题, 这时就需要借助程序调试(Debug)手段。

下面以习题 6 中给出的程序源代码为例, 简要说明程序调试方法。开始下列步骤前, 请读者参考实验 1.1.1 录入源代码, 并进行编译和链接。

1. 设置程序断点

调试的基本思想就是让程序运行时，在读者认为可能有错误的代码前暂停下来，然后，在人的控制下逐条运行语句，并通过在运行过程中查看相关变量的值，来判断错误产生的原因。想让程序运行到某一行前暂停下来，就需要将该行设成断点。设置断点的具体方法是在代码所在行的行首单击，如果断点设置成功，该行将被加亮显示（默认的加亮颜色是红色），如图 1-14 所示。图 1-14 中，将语句“z=x+y;”设置成断点，程序运行完语句“cout<<"x="<<x<<"y="<<y<<endl;”后，将会暂停。需要说明的是，一个程序中根据需要可以同时设置多个断点。

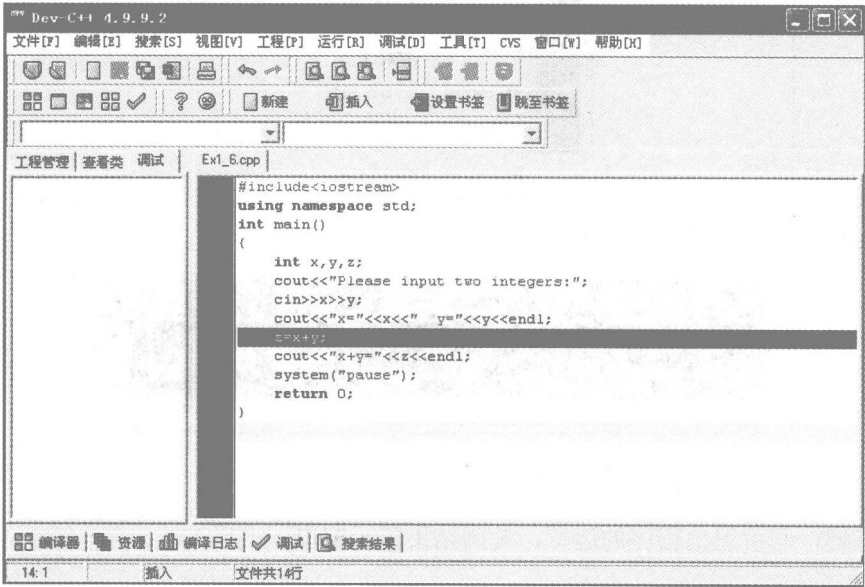


图 1-14 设置程序断点

如果想取消某个断点，只需在该代码行的行首再次单击。如果断点被取消，该行就不会被加亮显示。

2. 调试程序

正确设置断点后，如果程序已编译和链接通过，就可以对它进行调试。在 Dev-C++ 环境调试程序时，需要依次单击主菜单“调试”→“调试”菜单项（或者按快捷键〈F8〉）运行程序，如图 1-15 所示。

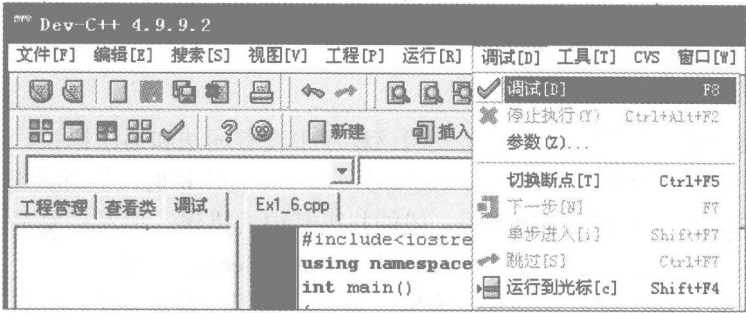


图 1-15 调试程序（一）

此时，程序将运行到第一个断点处暂停，并且该断点处的加亮色由红色变成蓝色，表示接下去将运行蓝色底色的代码，如图 1-16 所示。

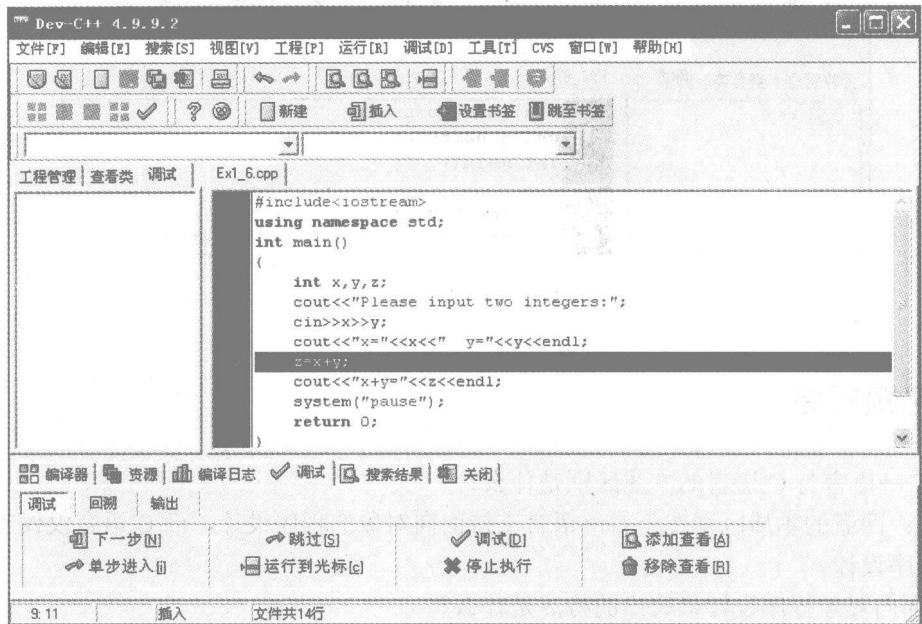


图 1-16 调试程序（二）

要想运行蓝色底色的代码，可以单击图 1-16 中的“下一步”、“单步进入”、“跳过”、“运行到光标”等按钮，或主菜单“调试”下的相应菜单项。其中，“下一步”的作用是运行下一行代码，如果下一行是对函数的调用，不进入函数体；“单步进入”的作用是运行下一行代码，如果下一行是对函数的调用，则进入函数体；“跳过”的作用是运行到下一个断点处；“运行到光标”的作用是运行到光标所在处。

3. 设置查看窗口

调试程序时，可能需要查看程序运行过程中变量的值，以检测程序对变量的处理是否正确。如果调试时需要查看某个变量的值，只需在调试过程中依次单击主菜单“调试”→“添加查看”菜单项，在弹出的“新变量”对话框中输入该变量的名称，然后单击“OK”按钮，如图 1-17 所示。新增的变量将会显示在最左边的“调试”选项卡中，如图 1-18 所示。如果“调试”选项卡不是当前页，可以单击使之成为当前页。

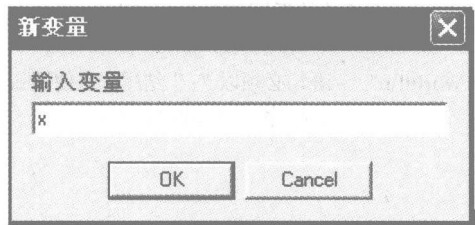


图 1-17 添加新的查看变量

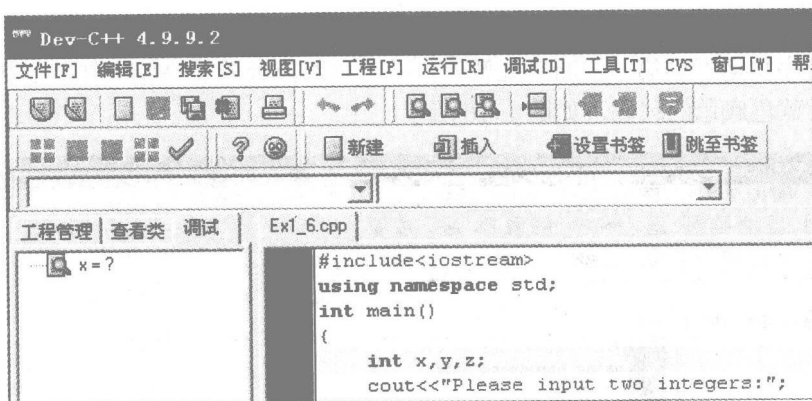


图 1-18 查看窗口

1.2 习题解答

1. C++语言与 C 语言的本质区别是什么？

解答：两者的本质区别在于 C++语言支持面向对象的程序设计，而 C 语言仅仅支持面向过程的程序设计。

2. 面向对象程序设计最突出的特点是什么？

解答：面向对象程序设计最突出的特点是封装性、继承性和多态性。

3. 开发一个 C++程序至少需要经过几个步骤？

解答：开发一个 C++程序至少需要经过编辑、编译、链接、运行 4 个步骤。

4. 找出下面程序中的错误，并改正，然后编译、链接、运行。它的运行结果是什么？

```
include<iostream.h>

void main()
{
    cout<<" Hello World!\n"
}
```

解答：改后如下（程序中注释部分对原有错误进行了说明）。

```
#include<iostream>           //原程序少“#”号，且标准 C++中，标准库头文件不应有扩展名.h

int main()                   //主函数的返回值类型只能是 int，不能是 void
{
    std::cout<<"Hello World!\n"; //语句必须以“;”结尾，且使用 cout 时，必须指定其所在的名空间
    return 0;                 //原程序缺少返回值语句
}
```

程序运行结果：

Hello World!