

21 世纪普通高等教育规划教材

C++ 语言

上机指导 实用教程

卢鹏丽 编著



机械工业出版社
CHINA MACHINE PRESS



21

21世纪中国计算机教育出版工程

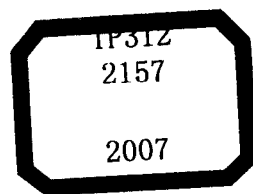
C++ 语言

上机指导 实用教程

卢毅 编著



清华大学出版社



21 世纪普通高等教育规划教材

C++ 语言上机指导实用教程

卢鹏丽 编著

张远平 审



机械工业出版社

C++ 语言是计算机及其相关专业重要的程序设计语言。本书作为目前高等学校普遍使用的 C++ 教材的辅助教材,通过一些经典的上机实例,使学生在掌握 C++ 知识的同时提高上机操作能力。全书共分 11 章,内容包括 C++ 语言基础知识、类和对象、静态成员、友元、构造函数和析构函数、运算符的重载、继承与派生、虚函数和多态性、模板、文件流等。本书还给出了与教材内容配套的多个上机实例,每个实例先给出其运行结果,然后是程序源代码,最后指出该程序在上机操作过程中的注意事项。重要的章节后有上机练习题及参考答案,以加深对知识的理解和掌握。

本书可作为大学计算机及相关专业的本科、专科生学习 C++ 程序设计的参考书,也可作为广大计算机爱好者的自学指导参考书。

图书在版编目 (CIP) 数据

C++ 语言上机指导实用教程/卢鹏丽编著. —北京:机械工业出版社, 2007.1

21 世纪普通高等教育规划教材

ISBN 7-111-20126-4

I .C... II .卢... 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 (2006) 第 124918 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:贡克勤 版式设计:冉晓华 责任校对:董纪丽

封面设计:姚毅 责任印制:杨曦

北京机工印刷厂印刷

2007 年 1 月第 1 版第 1 次印刷

184mm × 260mm · 11.5 印张 · 281 千字

定价:18.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68326294

编辑热线电话 (010) 88379711

封面无防伪标均为盗版

前 言

C++ 语言是在 C 语言的基础上发展起来的，不仅继承了 C 语言的原有特点，而且引入了面向对象的程序设计方法，是当今计算机专业及其相关专业的大学生所必须掌握的程序设计语言之一。

很多初学者都会提出一个共同的问题，即怎样才能学好 C++？其实，没有什么捷径可走，惟有自己动手多编写程序。只有多编写程序和在上机调试程序，才能把 C++ 的语法规则、编程的基本方法和技巧、各种算法变成实实在在的知识。

目前市场上有关 C++ 面向对象程序设计的书很多，但如何指导学生上机操作的书却很少，学生往往在上机时不知道干什么，或者遇到问题时无法解决。针对以上问题，我们编写了这本《C++ 语言上机指导实用教程》。本书作为目前高等学校普遍使用的 C++ 教材的辅助教材，每节给出与教材内容配套的多个上机实例。这些实例经过反复的挑选，既有利于学生掌握有关知识，又不失趣味性。每个实例先给出其运行结果，然后是程序源代码，最后指出该程序在上机操作过程中的注意事项，使学生在掌握 C++ 知识的同时提高上机操作能力。而且在重要的章节后有上机练习题及参考答案，以锻炼学生的能力，加深对知识的理解和掌握。

另外，很多学习 C++ 语言的学生都已经学完了 C 语言，因此本书主要介绍 C++ 语言独有的知识，而不再介绍与 C 语言共同的部分，如常量、变量、表达式、运算符、数组、顺序、选择、循环结构、指针、函数、结构体等，目的是重点突出 C++ 语言，使读者有重点地掌握 C++ 语言。

全书由卢鹏丽编写，张远平副教授审定。

由于时间仓促，加之作者水平有限，不当之处在所难免，敬请读者批评指正。

编 者
2006 年 8 月

目 录

前言

第 1 章 预备知识	1
1.1 简单的输入/输出	1
1.2 Visual C++ 6.0 的操作环境	4
1.3 程序调试——Debug 功能	10
第 2 章 C++ 语言基础	14
2.1 内嵌（联）函数	14
2.2 函数的重载	16
2.3 带默认参数值的函数	19
2.4 const 类型限定符	21
2.5 new 与 delete	23
第 3 章 类和对象	28
3.1 类的定义	28
3.2 数据成员及类成员的访问控制	33
3.3 对象及指向对象的指针	37
3.4 类与结构的关系	42
3.5 类定义与头文件	43
3.6 上机练习题	47
3.7 上机练习题参考答案	47
第 4 章 构造函数和析构函数	56
4.1 构造函数	56
4.2 析构函数	64
4.3 默认构造函数与拷贝构造函数	70
4.4 构造函数的重载	74
4.5 一个类的对象作为另一个类的数 据成员	76
4.6 上机练习题	79
4.7 上机练习题参考答案	79
第 5 章 静态成员	88
5.1 静态数据成员	88
5.2 静态成员函数	91

第 6 章 友元	97
6.1 友元函数	97
6.2 友元类	99
6.3 上机练习题	104
6.4 上机练习题参考答案	104
第 7 章 this 指针与运算符的重载	110
7.1 this 指针	110
7.2 运算符的重载	114
7.3 上机练习题	119
7.4 上机练习题参考答案	119
第 8 章 继承与派生	125
8.1 单继承	125
8.2 多继承	132
8.3 虚基类	134
8.4 上机练习题	138
8.5 上机练习题参考答案	138
第 9 章 虚函数和多态性	145
9.1 虚函数和多态性概述	145
9.2 纯虚函数和抽象类	150
第 10 章 模板	153
10.1 函数模板	153
10.2 类模板	158
10.3 上机练习题	160
10.4 上机练习题参考答案	160
第 11 章 I/O 流	166
11.1 文件流	166
11.2 上机练习题	174
11.3 上机练习题参考答案	174
参考文献	178

第1章 预备知识

C++ 语言是在 C 语言的基础上发展起来的，它不仅继承了 C 语言简洁、高效的特点，而且引入了面向对象程序设计的方法，是一种过程性和对象性相结合的程序设计语言。本章主要介绍简单的输入/输出，在 Visual C++ 6.0 环境下编辑、编译、运行简单的程序，设置断点、调试程序、观察程序运行过程中变量的当前值的方法。

1.1 简单的输入/输出

C++ 中的输入/输出功能是由 I/O 流来实现的。C++ 提供了两个标准的 I/O 流，即流 cin 和流 cout。cin 实现输入操作，即从键盘上输入；cout 实现输出操作，即由屏幕输出。

本节主要介绍 C++ 无格式输入输出 “cout <<”、“cin >>” 及其与 C 语言的格式输入输出 scanf()、printf() 的区别。

标准输出的格式为：cout << 表达式 << 表达式…

标准输入的格式为：cin >> 表达式 >> 表达式…

练习一 在屏幕上输出一行简单文字。

输出结果

Hello, world!

Press any key to continue

程序源代码

```
#include "stdafx.h"
#include "iostream.h"
void main (void)
{   cout << "Hello, world! ";}
```

注意事项

- C 语言的输入输出：scanf、printf 都是格式输入输出。
- C++ 中：可以用格式输入输出 scanf、printf，也可以用无格式输入输出流操作符。
- 流操作符 cin、cout 与 printf、scanf 的区别：
 - 1) 头文件为 iostream.h 而不是 stdio.h。
 - 2) 不需要格式指定符。
 - 3) cin 中没有对操作数使用地址操作符 (&)，而 scanf 则必须用。

练习二 在屏幕上输出变量的值，用格式控制函数 `setw (int)` 来设置输出数据间的宽度。

输出结果

2 3 4

3 5 7

Press any key to continue

程序源代码

```
#include "stdafx.h"
#include "iostream.h"
#include "iomanip.h"
void main (void)
{
    int a = 2, b = 3, c = 4;
    cout << a << setw (3) << b << setw (6) << c << endl; //控制输出格式
    cout << a + 1 << setw (3) << b + 2 << setw (6) << c + 3 << endl;
}
```

注意事项

- `setw ( )` 包含在头文件 `iomanip.h` 中。
- C++ 用 “`endl`” 打一个回车，相当于 c 中的 “`\n`”，而 C++ 中两种方式都支持。

C++ 的注释有两种形式：

单行注释 //

多行注释 /* */

练习三 从键盘上输入两个数，输出两数之和。

输出结果

输入第一个数 : 4

输入第二个数 : 5

a + b = 9

Press any key to continue

程序源代码

```
#include "stdafx.h"
#include "iostream.h"
int main (int argc, char * argv [])
{
    double a, b;
    cout << "输入第一个数 :"; cin >> a;
    cout << "输入第二个数 :"; cin >> b;
    cout << "a + b = " << a + b << endl;
    return 0;
}
```

```
}

```

注意事项

- `cin >> a;` 输入单个数后以回车结束。
- 若 `cin >>` 后面跟多个表达式, 如 `cin >> a1 >> a2 >> a3`, 则要输入多个数据。输入值之间以空格隔开, 输入结束后以回车结束。

练习四 在屏幕上输出各种数据类型的存储长度。

输出结果

整型 32

短整型 16

长整型 32

无符号整型 32

单精度实型 32

双精度实型 64

字符型 8

Press any key to continue

程序源代码

```
#include "stdafx.h"
#include "iostream.h"

void main (void)
{
    int n;
    short s;
    long l;
    unsigned int ul;
    float f;
    double d;
    char c;

    cout << "整型 " << 8 * sizeof (n) << endl;
    cout << "短整型 " << 8 * sizeof (s) << endl;
    cout << "长整型 " << 8 * sizeof (l) << endl;
    cout << "无符号整型 " << 8 * sizeof (ul) << endl;
    cout << "单精度实型 " << 8 * sizeof (f) << endl;
    cout << "双精度实型" << 8 * sizeof (d) << endl;
    cout << "字符型 " << 8 * sizeof (c) << endl;
}
```

注意事项

C++ 中的数据类型除了上面的基本类型外, 还有构造类型 (数组、结构体、共用体、枚举、类) 和指针类型。

练习五 printf ()、scanf () 函数的使用。

输出结果

i, j =

2, 3

i = 2, j = 3 Press any key to continue

程序源代码

```
#include "stdafx.h"
#include "stdio.h"
int main (int argc, char * argv [])
{
    inti, j;
    printf ("i, j = \n");
    scanf ("%d,%d", &i, &j);
    printf ("i = %d, j = %d", i, j);
    return 0;
}
```

注意事项

- C++ 中：可以用格式输入输出 scanf、printf，也可以用无格式输入输出流操作符。
- printf、scanf 头文件为 stdio.h。scanf 要对操作数使用地址操作符 (&)。

1.2 Visual C++ 6.0 的操作环境

C++ 的编译环境有 Turbo C++、Borland C++、VC++，大家可以在任意一种环境下编译。本书所有的程序都已在 Visual C++ 6.0 环境下编译通过。

本节介绍 Visual C++ 6.0 的操作环境，在 Visual C++ 6.0 环境下进行 C++ 源代码的编辑、编译、运行。

1. Visual C++ 6.0 介绍

Microsoft 公司出版的 Visual C++ 6.0 是一套可视化的 C++ 程序编辑软件，环境与一般的编辑软件相似，包含一个简单的编辑区、一个功能齐全的工具栏及菜单。

使用 Visual C++ 6.0 前，首先要将其安装到用户的计算机上（安装过程略）。

2. 进入 Visual C++ 6.0

启动 Visual C++ 6.0 的方法：

- 如果已经把 Visual C++ 6.0 创建成了桌面快捷方式，则双击图标即可进入 Visual C++ 6.0 系统。
- 另一种方法可以从 [开始] / [程序] / [Microsoft Visual Studio6.0] / [Microsoft Visual C++ 6.0] 即可进入 Visual C++ 6.0 系统。

3. Visual C++ 6.0 的工作窗口

启动 Visual C++ 6.0 后，就会出现 Visual C++ 6.0 窗口。图 1-1 为一个小程序编辑后的

窗口状态。可以看到，Visual C++ 6.0 窗口包括以下内容：

- ① 菜单栏：包括文件、编辑、视图、插入、工程、编译、工具、窗口、帮助主菜单，每个主菜单下有相应的子菜单。
- ② 标准工具栏：包括打开文件、保存文件等常用的操作。
- ③ 向导栏：可列出程序中含有的类以及生成类向导。
- ④ 建立程序工具栏：包括 Compile（编译程序）、Build（建立可执行文件）、BuildStop（停止建立可执行文件）、BuildExecute（执行程序）、Go（执行程序到下一个断点处）、Insert/Remove Breakpoint（插入/删除程序的断点）。
- ⑤ 程序编辑区：编写程序的区域。
- ⑥ 输出窗口：显示程序编译期间产生的输出信息，如 error（错误）warning（警告）等信息。
- ⑦ 信息显示区：显示编辑程序或执行功能时的信息。
- ⑧ Workspace（工作空间）窗口：其底端显示两个标签，即 ClassView 和 FileView，可以单击，分别打开。

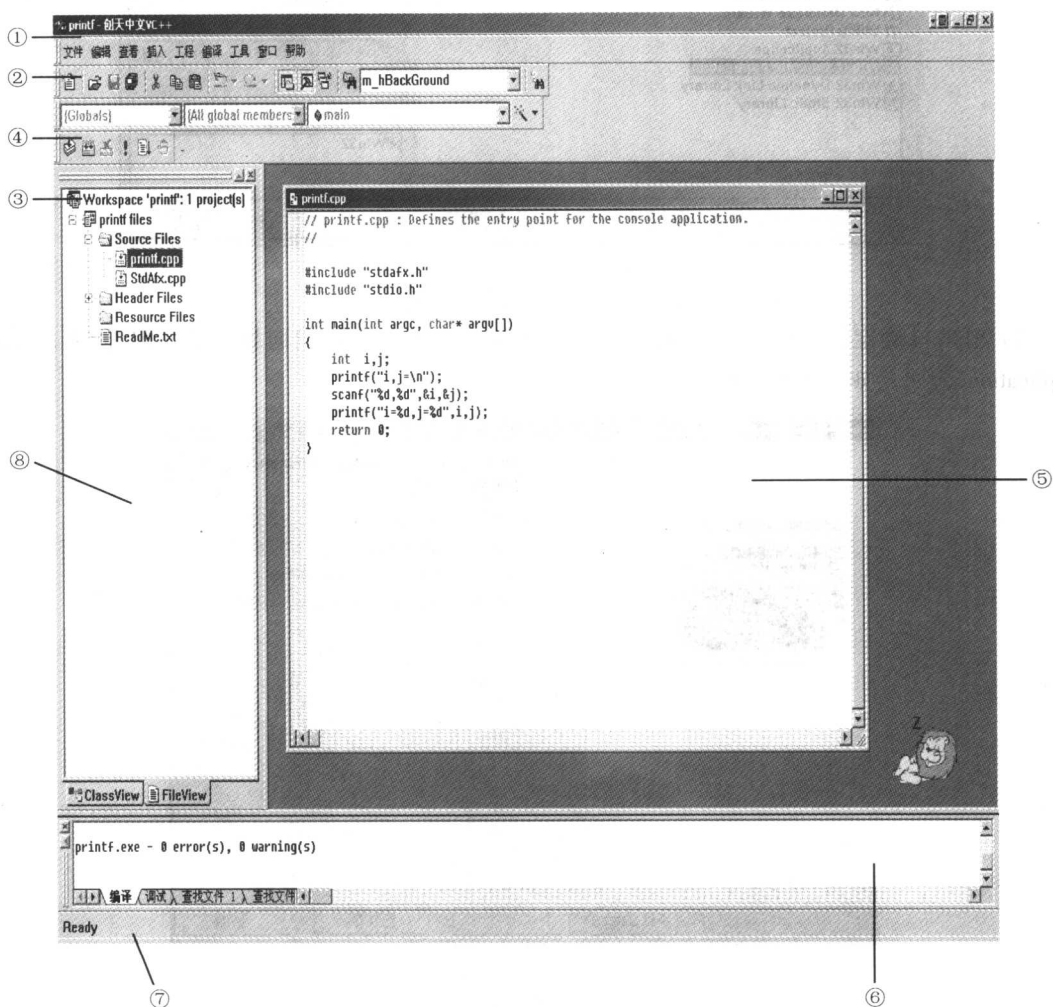


图 1-1 窗口状态

4. 使用 Visual C++ 6.0

(1) 编辑一个新文件

第一步：用 AppWizard 创建一个控制台应用程序。

1) 执行【文件】→【新建】菜单命令，显示出“新建”对话框。选择其中的【工程】标签，从列表框中选择 Win32 Console Application 项。在工程名框中键入控制台应用程序的项目名称，此项目我们取名为 user1。在位置框中键入此项目所在的文件夹，如图 1-2 所示。

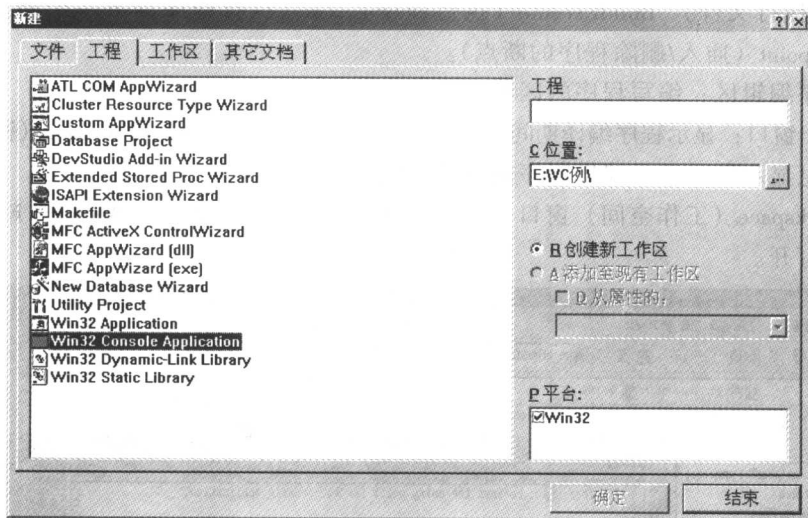


图 1-2 “新建”对话框

2) 单击【确定】按钮。接下来显示一个询问项目类型的应用向导，选中其中的 A simple application 选项，如图 1-3 所示。

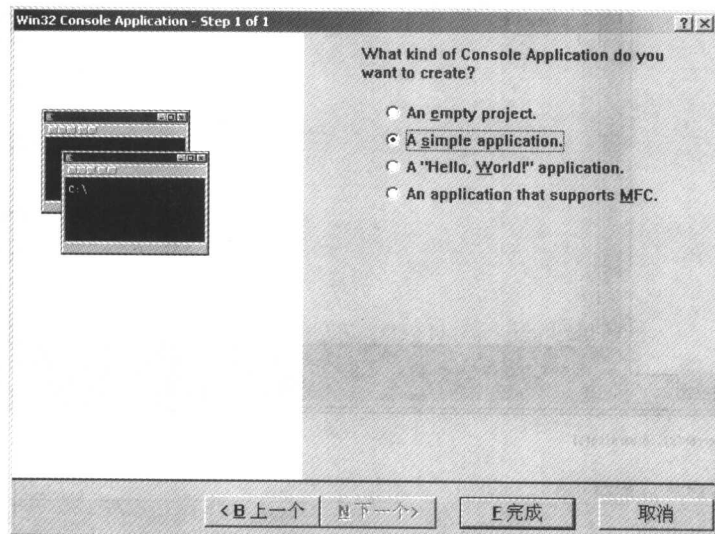


图 1-3 A simple application 选项

3) 单击【完成】按钮，将出现“新建工程信息”对话框，如图 1-4 所示，单击【确定】

按钮，系统将自动创建此应用程序。

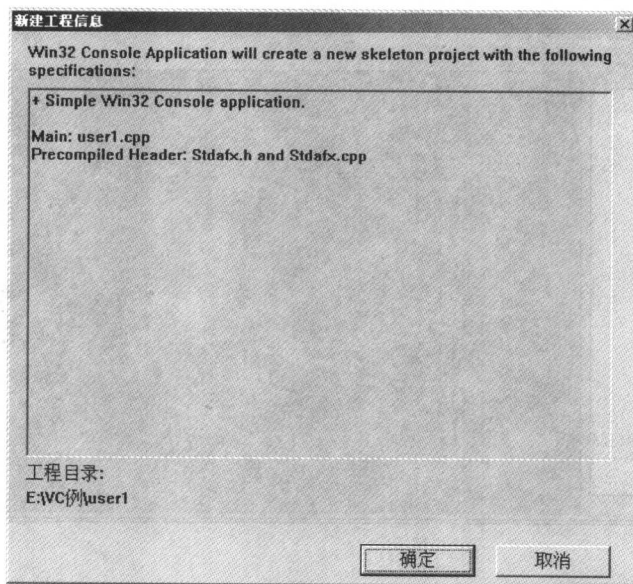


图 1-4 新建工程信息对话框

第二步：打开 user1.cpp 文件。

通过上述过程，创建好了一个控制台应用程序 user1 的程序框架。在项目工作区窗口中选择 ClassView 标签项，可以看到生成的类信息，如图 1-5 所示。选择 FileView 标签项，可以看到 AppWizard 生成了 user1.cpp, Stdafx.cpp, Stdafx.h, ReadMe, txt 几个文件，如图 1-6 所示。其中 user1.cpp 是 AppWizard 产生的程序源代码文件，用户几乎所有的代码都是添加在这个文件中的。

第三步：添加程序源代码。

上述源代码只是一个框架，用户必须添加一些代码，以实现自己需要的功能，代码的添加过程如下。

1) 在项目工作区窗口中选择 FileView 标签项，单击 user1.cpp 源文件，程序编辑区中将出现一些源代码，如图 1-7 所示。

2) 在程序编辑区中添入自己的程序代码，如图 1-8 所示。

(2) 保存文件 利用【文件】菜单中的 Save all 命令项或标准工具栏的 Save all 按钮将所有文件保存。

(3) 关闭/打开一个已存在的文件

1) 打开【文件】菜单，单击选中【关闭工作区】菜单，关闭工作空间。

2) 打开【文件】菜单，单击选中【打开工作区】菜单，选择搜索路径，选择要打开的工作区（此处可以选择 user1.dsw），单击【打开】即可对此文件进行编辑、编译、运行操作。

(4) 编译和构建 编辑好源程序后，即可以对程序进行编译、构建。

1) 打开【编译】菜单，选择其中的 Compile 编译器（此处选择 [编译 user1.cpp]）即可对程序进行编译，生成 *.obj 文件。编译之后，在输出窗口中会输出 error 和 warning 的相关信息。移动光标到此窗口，上下移动光标察看错误信息，并且修改源程序后重新编译，直到出

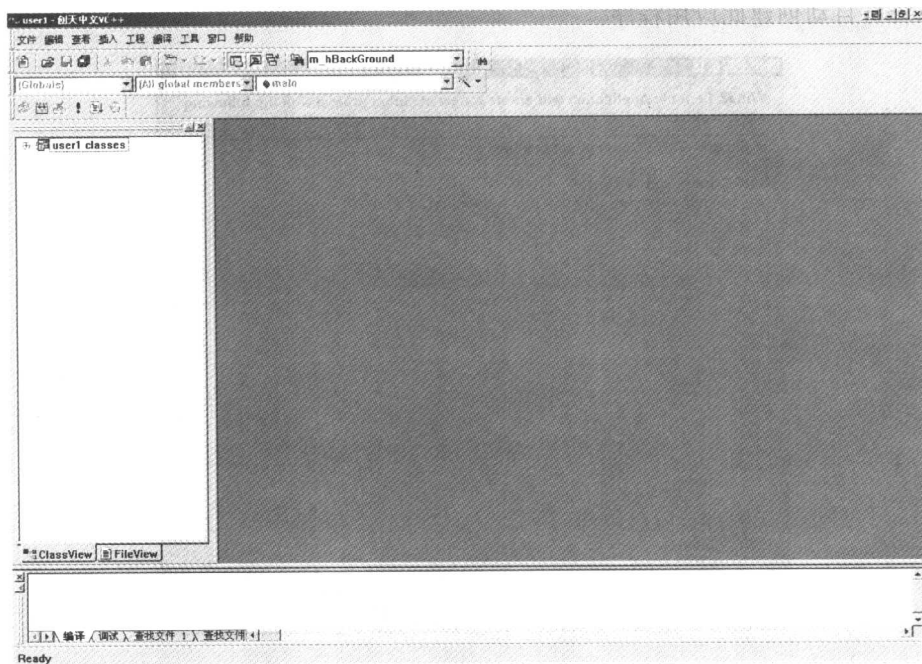


图 1-5 类信息

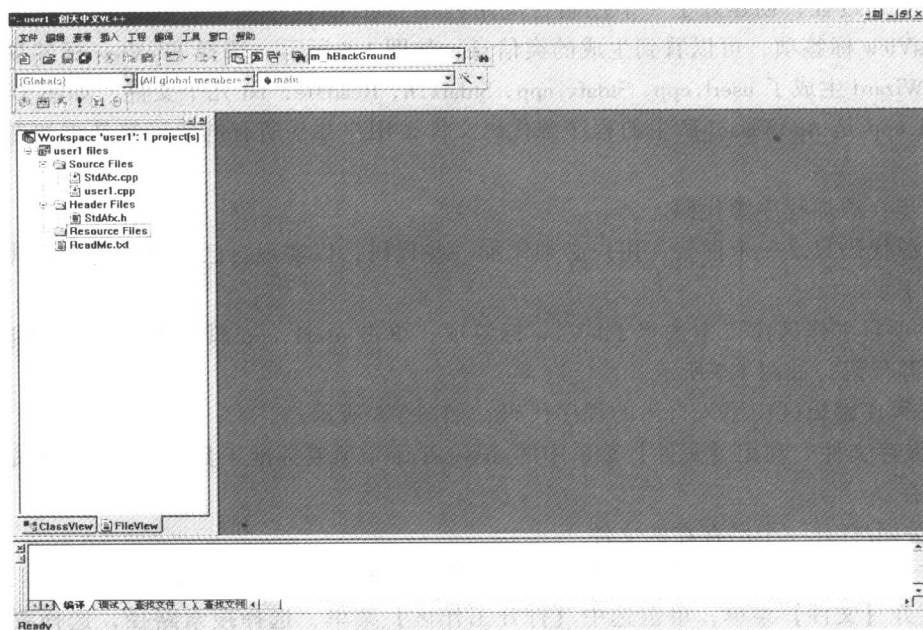


图 1-6 user1. cpp, Stdafx. cpp, Stdafx. h, ReadMe, txt 文件

现“error 0”为止，编译完成。

2) 打开【编译】菜单，选择【构建】（此处可以选择[构建 user1.exe]），对程序进行构建，生成可执行文件*.exe。

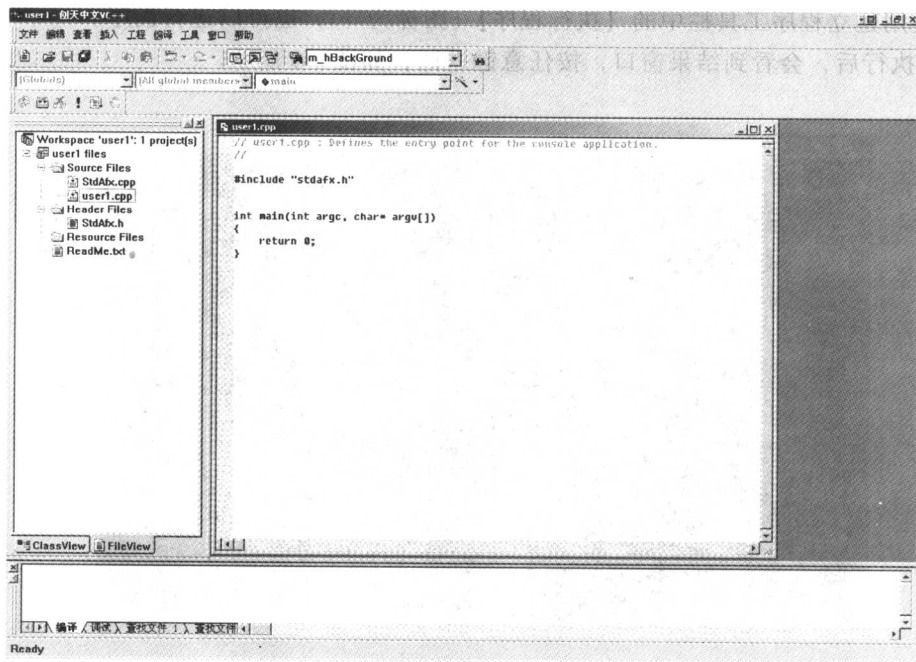


图 1-7 源代码

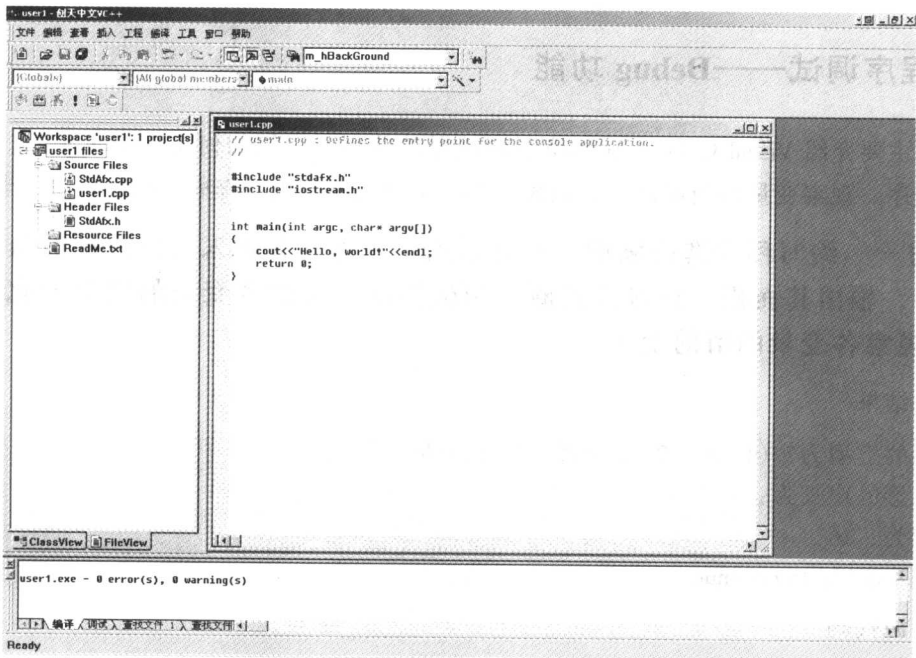


图 1-8 程序代码

(5) 运行 执行过编译和构建之后，生成了可执行文件，即可运行程序了。

1) 打开【编译】菜单，选择【执行程序】（此处可以选择 [执行 user1.exe]），即可运行程序。

2) 使用建立程序工具栏中的【执行程序】(图标为 !), 也可以运行程序。程序执行后, 会看到结果窗口, 按任意键返回, 如图 1-9 所示。

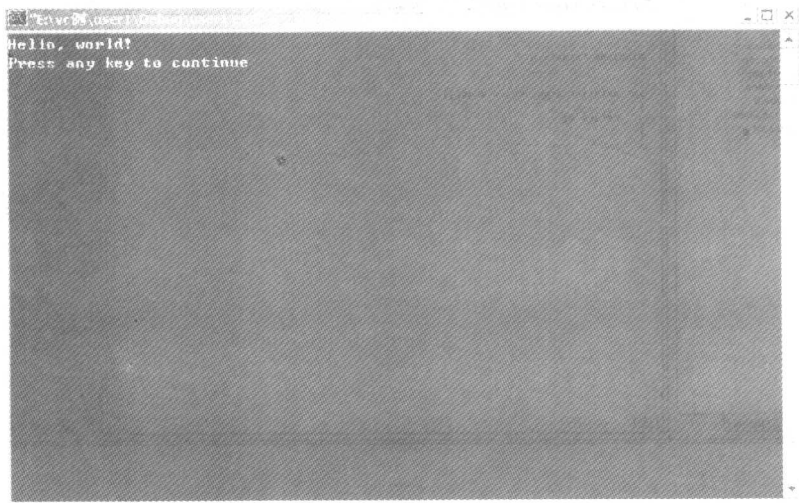


图 1-9 结果窗口

1.3 程序调试——Debug 功能

本节主要介绍 Visual C++ 6.0 环境下调试器的使用。介绍如何实现单步跟踪、设置断点调试程序、观察变量的当前值等, 使我们可以跟踪程序的执行流程, 观察变量的当前值。

练习一 编写程序选择圆形、长方形或正方形, 并输入其相应的数值 (半径或边长), 输出其面积。练习设置断点调试程序, 观察各变量的值的变化; 单步执行, 观察各变量的值的变化。

输出结果

图形的类型为? (1. 圆, 2. 长方形, 3. 正方形) 3

正方形的边长为: 4

面积为: 16

Press any key to continue

程序源代码

```
#include "stdafx.h"
#include "iostream.h"
const float PI = 3.14;
int main (int argc, char * argv [])
{
    int i;
```

```
float radius, a, b, area;
cout << "图形的类型为? (1. 圆, 2. 长方形, 3. 正方形)";
cin >> i;
switch (i)
{
    case 1:
        cout << "圆形的半径为";
        cin >> radius;
        area = PI * radius * radius;
        cout << "面积为:" << area << endl;
        break;
    case 2:
        cout << "长方形的长为:";
        cin >> a;
        cout << "长方形的宽为:";
        cin >> b;
        area = a * b;
        cout << "面积为:" << area << endl;
        break;
    case 3:
        cout << "正方形的边长为:";
        cin >> a;
        area = a * a;
        cout << "面积为:" << area << endl;
        break;
    default: cout << "非法输入!" << endl;
}
return 0;
}
```

调试程序的方法:

1) 在行“cout << “图形的类型为? (1. 圆, 2. 长方形, 3. 正方形);”处设置断点。在此行单击右键, 会出现一个菜单, 如图 1-10 所示。

2) 选择 Insert/Remove Breakpoint 选项, 此行左边会出现一个圆点, 代表此行已设了一个断点。

3) 选择菜单命令 Build/Start Debug /Go, 或按下快捷键 F5, 系统进入 Debug 调试状态, 程序开始运行, 一个 DOS 窗口出现, 程序暂停在断点处。此时的界面如图 1-11 所示。

4) 从左下方的 Variables 窗口中可以观察各变量的当前值, 也可以在右下方的 Watch 窗口中输入想要跟踪的变量名, 如 a、b、area, 可以观察各变量的值的变化。此时如果找不到这两个窗口, 可以在菜单【查看】/【调试窗口】下选中 Watch 和 Variables, 显示这两个窗口。