

全国高职高专规划教材

面向对象的程序设计 C++实训教程

Object-Oriented Programming
with C++ in Practice

杨学明 主 编
刘加海 余建军 副主编



科学出版社

www.sciencep.com



全国高职高专规划教材

面向对象的程序设计

C++实训教程

杨学明 主 编

刘加海 余建军 副主编

科 学 出 版 社

北 京

内 容 简 介

本书为《面向对象的程序设计 C++》的配套教材。总共设计了 18 个实训题目,并在每个实训里设有小实验,在内容安排上基本与教材《面向对象的程序设计 C++》相一致。每个实训都有内容提示,实训目标、实训任务、实训步骤等内容,后面还配有相应的习题,学生可以在参照例题的基础上独立完成。此外,本书还带有三个附录,附录 1 是“ASCII 码表”;附录 2 “综合实例”给出了几个实用的系统,更加紧密地联系到了实际;附录 3 “C++语言编程规范”能引导学生编写高质量的代码。

本书适合作为高等院校相关专业的本科、专科、高职学生学习 C++ 程序设计的教材以及学习参考书。

图书在版编目(CIP)数据

面向对象的程序设计 C++实训教程/杨学明主编. —北京:科学出版社, 2003

(全国高职高专规划教材)

ISBN 7-03-011983-5

I. 面... II. 杨... III. C 语言—程序设计—高等学校—教学参考书
IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 065820 号

策划编辑:李振格/责任编辑:万国清

责任印制:吕春珉/封面设计:东方人华平面设计部

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社总发行 各地新华书店经销

*

2003 年 8 月第 一 版 开本:787×1092 1/16

2003 年 8 月第一次印刷 印张:12 1/4

印数:1—5 000 字数:274 000

定价:18.00 元

(如有印装质量问题,我社负责调换<长虹>)

全国高职高专规划教材编委会名单

主 任 俞瑞钊

副主任 陈庆章 蒋联海 周必水 刘加海

委 员 （以姓氏笔画为序）

王 雷 王筱慧 方 程 方锦明 卢菊洪 代绍庆
吕何新 朱 炜 刘向荣 江爱民 江锦祥 孙光弟
李天真 李永平 李良财 李明钧 李益明 余根墀
汪志达 沈凤池 沈安衢 张 元 张学辉 张锦祥
张德发 陈月波 陈晓燕 邵应珍 范剑波 欧阳江林
周国民 周建阳 赵小明 胡海影 秦学礼 徐文杰
凌 彦 曹哲新 戚海燕 龚祥国 章剑林 蒋黎红
董方武 鲁俊生 谢 川 谢晓飞 楼 丰 楼程伟
鞠洪尧

秘书长 熊盛新

本书编写人员名单

主 编 杨学明

副主编 刘加海 余建军

撰稿人 史宏杰 杨学明 刘加海 余建军

前 言

面向对象程序设计语言 C++具有强大的功能，为程序设计者提供了良好的程序开发环境，可产生模块化程度高，重用性和可维护性好的程序。目前 C++语言已经在各个领域得到了广泛的应用，能适用于设计各类应用程序和系统程序，特别适合中等和大型程序的开发项目。

C++程序设计语言是一门实践性很强的课程，仅仅通过阅读教科书或听课是不可能掌握的。学习程序设计最重要的环节就是实践。

在多年从事 C 和 C++程序设计语言的教学过程中，编者体会最深的是，要求学生理解和掌握程序设计语言的语法和语义是比较容易的，但要求学生掌握程序设计语言的基本思想和基本方法是不容易的。相信每个讲授程序设计语言的教师都有过这样的经历：有些问题尽管我们在课堂上再三强调，反复举例，学生还是不能够完全理解，上机时错误百出。根据多年的教学实践，我们编写本书的主要目的就是帮助学生解决这一问题，使学生能从看懂教科书过渡到具有基本的编程能力。

本书作为与《面向对象的程序设计 C++》配套的实训教材，在内容安排上基本与教材《面向对象的程序设计 C++》相一致。全书总共精心设计了 18 个实训题目，并在每个实训里设有小实验。另外本书还有 3 个附录。附录 1 “ASCII 码表”，方便学生在进行编程时查询 ASCII 码值。附录 2 “综合实例”，提供了 5 个具有实用意义的参考程序，学生可以作为参照。附录 3 “C++语言编程规范”，指导学生进行规范化的程序设计，养成良好的编程习惯。

由于编者水平有限，书中难免会有不妥和错误之处，恳请读者不吝赐教。

编 者
2003 年 7 月

目 录

第 1 章 Visual C++ 6.0 开发环境应用入门	1
1.1 Visual C++ 6.0 源程序编辑、编译、连接、运行过程	1
1.2 Visual C++ 6.0 常用工具栏	7
1.3 Visual C++ 6.0 中常用菜单	8
1.4 打开已存在的文件	9
1.5 C++程序的调试	10
第 2 章 函数的应用	16
2.1 函数的调用	16
2.2 内联函数示例	18
2.3 函数重载示例	21
2.4 带缺省形参值的函数示例	23
习题	30
第 3 章 内存管理和数据保护	31
3.1 使用 new 运算符和 delete 运算符管理内存	31
3.2 使用 malloc()函数和 free()函数管理内存	33
3.3 使用 const 关键字来保护数据	36
习题	40
第 4 章 理解类与对象	41
4.1 类与对象	41
4.2 静态成员	44
4.3 友元函数	46
4.4 容器类	48
习题	51
第 5 章 继承和派生类的应用	53
5.1 单继承和多重继承	53
5.1.1 单继承示例	53
5.1.2 多重继承示例	56
5.2 单继承中的访问控制	59
5.2.1 公有继承示例	59
5.2.2 私有继承示例	60
5.2.3 保护继承示例	61
5.3 继承中的构造函数和析构函数	62
5.4 多重继承中的二义性	65

5.5 虚基类	67
习题	68
第 6 章 运算符重载的应用	69
6.1 成员函数和友元函数的重载	69
6.1.1 运算符重载为成员函数示例	69
6.1.2 运算符重载为友元函数示例	71
6.2 字符串运算中的运算符重载	72
6.3 二元运算符的重载	74
习题	76
第 7 章 转换函数的应用	77
7.1 基本数据类型向对象类型的转换	77
7.2 对象类型向基本数据类型的转换	79
7.3 不同类的对象互相转换	82
习题	85
第 8 章 引用的应用	86
8.1 对象的引用	86
8.2 引用和 const 的连用	89
8.3 函数调用过程中的值传递和地址传递	91
习题	95
第 9 章 拷贝构造函数的应用	97
9.1 拷贝构造函数	97
9.2 拷贝构造函数和重载的赋值运算符	99
习题	101
第 10 章 多态性的理解	102
习题	103
第 11 章 虚函数的应用	104
11.1 虚函数	104
11.2 动态绑定	107
习题	108
第 12 章 虚析构函数的应用	109
12.1 使用非虚析构函数的基类示例	109
12.2 使用虚析构函数的基类示例	111
习题	112
第 13 章 抽象类的应用	113
13.1 纯虚函数	113
13.2 抽象类	115
习题	116
第 14 章 模板的应用	117
14.1 函数模板	117

14.2 类模板	120
习题	122
第 15 章 异常处理的应用	123
15.1 C++异常处理的实现	123
15.1.1 异常处理的概念	123
15.1.2 C++异常处理示例	123
15.2 异常处理中的构造与析构	125
习题	126
第 16 章 用 iso 类的成员函数实现格式化输入输出	127
16.1 左右对齐示例	127
16.2 设置填充字符示例	128
16.3 设置及清除格式标志示例	129
16.4 流错误状态示例	130
习题	132
第 17 章 用操纵符实现格式化输入输出	133
17.1 整数的格式化输入输出	133
17.1.1 十进制、八进制、十六进制整数的输出示例	133
17.1.2 十进制、八进制、十六进制整数的输入示例	134
17.2 浮点数精度和域宽的使用	135
17.2.1 设置浮点数精度示例	135
17.2.2 域宽的使用示例	136
习题	136
第 18 章 文件读写的应用	137
18.1 顺序访问文件的使用	137
18.1.1 格式化文件	137
18.1.2 顺序访问文件示例	137
18.2 随机访问文件的使用	138
18.2.1 二进制文件	138
18.2.2 随机访问文件示例	138
习题	142
附录	143
附录 1 ASCII 码表	143
附录 2 综合实例	143
附录 3 C++语言编程规范	175
主要参考文献	185

第 1 章 Visual C++ 6.0 开发环境

应用入门

实训目的

- 掌握在 Visual C++ 6.0 的集成环境下源程序的编辑
- 掌握通过项目来编译和连接程序
- 熟悉 Visual C++ 6.0 的集成环境
- 了解 Visual C++ 6.0 环境上机中的常见错误提示及修改方法

实训内容

● 本章主要介绍 Visual C++ 6.0 的集成环境下源程序的编辑、编译和连接的过程，熟悉 Visual C++ 6.0 的集成环境，包括工具栏、菜单的操作方法，以及在上机操作过程中的常见错误以及解决的方法。

● Visual C++ 6.0 提供了一个支持可视化编程的集成开发环境：Visual Studio（或 Developer Studio）。Developer Studio 是一个通用的应用程序集成开发环境，它包含了一个文本编辑器、资源编辑器、工程编译工具、一个增量连接器、源代码浏览器、集成调试工具，以及一套联机文档。Developer Studio 把所有的 Visual C++ 工具结合在一起，集成为一个整体，通过一个由窗口、对话框、菜单、工具栏、快捷键及宏组成的和谐系统，可以观察和控制整个开发进程。

1.1 Visual C++ 6.0 源程序编辑、编译、连接、运行过程

为了启动 Visual C++ 6.0 的开发平台，执行过程为：从 Windows 的“开始”菜单，选择“程序”→“Microsoft Visual Studio”→“Microsoft Visual C++ 6.0”，用鼠标左键单击“Microsoft Visual C++ 6.0”，如图 1.1 所示，弹出如图 1.2 所示的提示框，如不需要进一步了解，可单击“C 结束”。

随后会进入 Developer Studio 的应用程序集成开发环境，如图 1.3 所示的界面。

用鼠标左键单击“File”→“新建”命令，得到“新建”对话框，如图 1.4 所示。

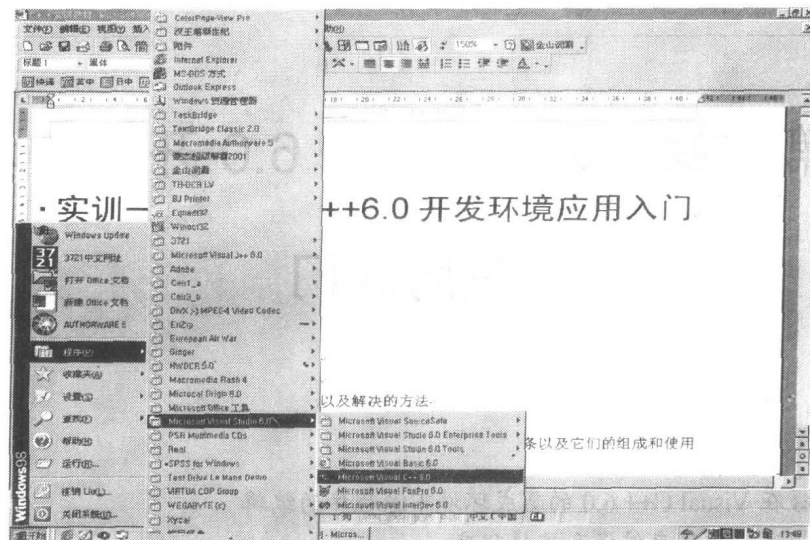


图 1.1 从 Windows 开始菜单进入 Visual C++ 6.0

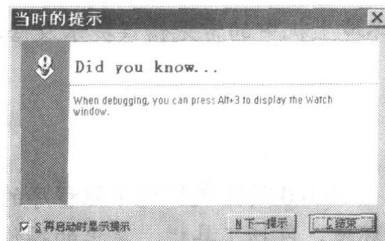


图 1.2 单击命令按钮“C 结束”

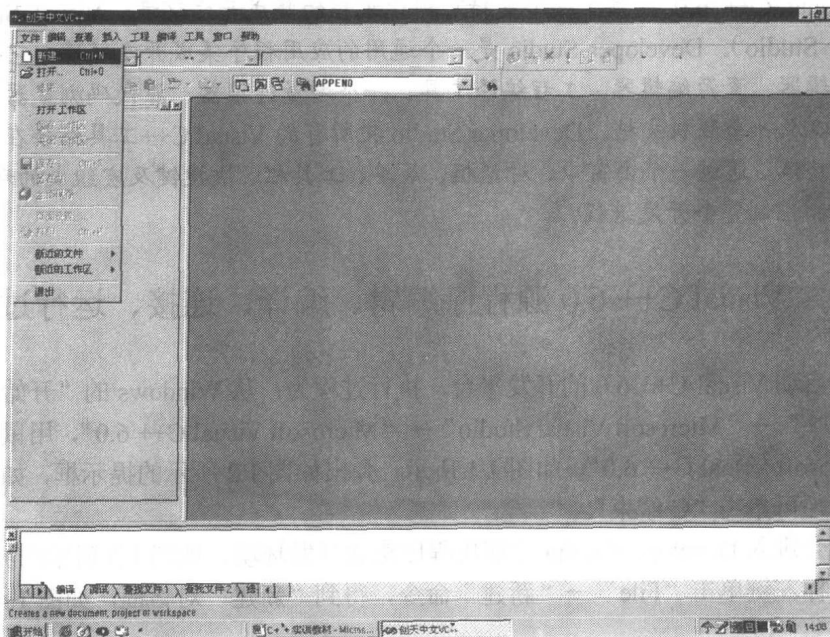


图 1.3 Developer Studio 的应用程序集成开发环境

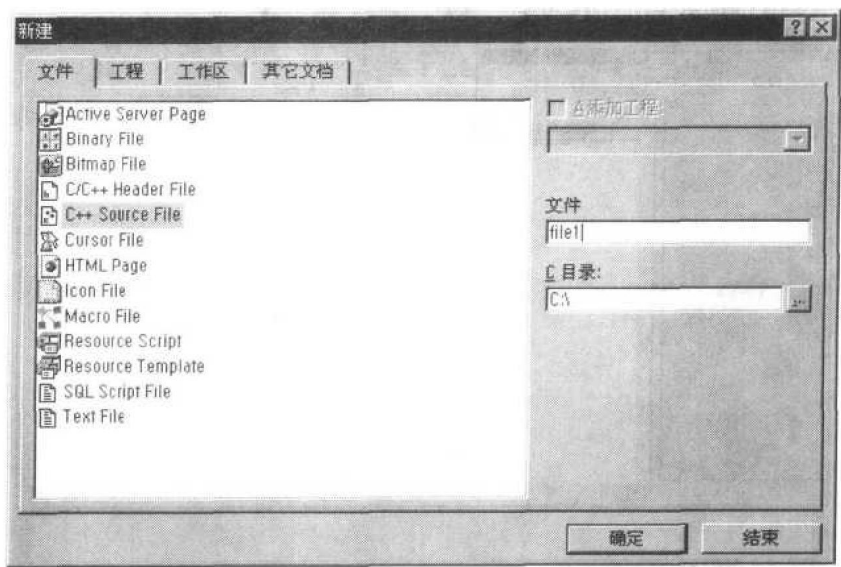


图 1.4 “新建”对话框

在“新建”对话框中单击“文件”标签，选择标签“C++ Source File”，并且在右边“文件”文本框中输入文件名：file1，存盘时系统会自动加上扩展名。此时完整的文件名为 file1.cpp，单击命令按钮“确定”，出现如图 1.5 所示的源程序编辑区。在编辑区光标处开始输入源程序，输入的过程中，完全可以用 Microsoft Word 的方法对文字进行插入、删除、修改、复制、粘贴、移动等操作。源程序编辑完毕，如图 1.6 所示，单击菜单栏的菜单“编译”→“编译 file1.cpp”，出现如图 1.7 所示的对话框。编译命令需要一个活动工作区，是否建立一个默认的工作区，单击“是(Y)”命令，建立一个默认的工作区。在图 1.8 中将改动保存到文件 C:\file1.cpp 中。

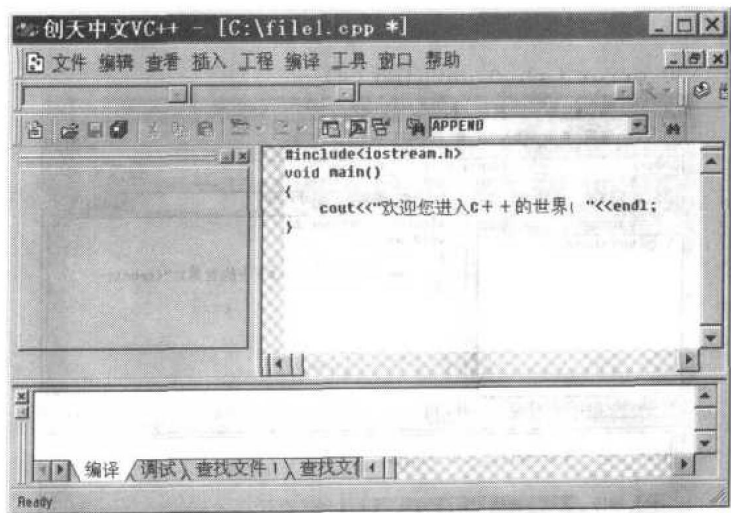


图 1.5 源程序编辑区

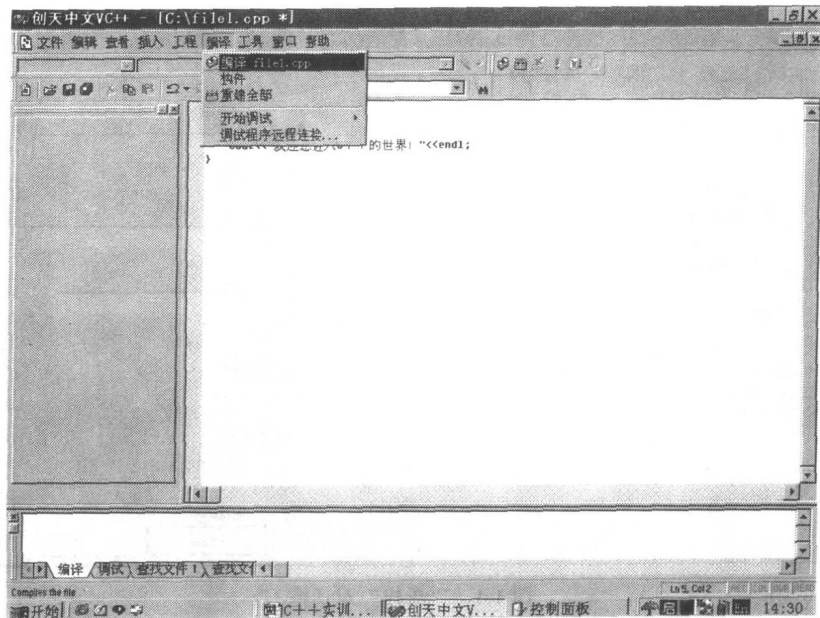


图 1.6 编译源程序

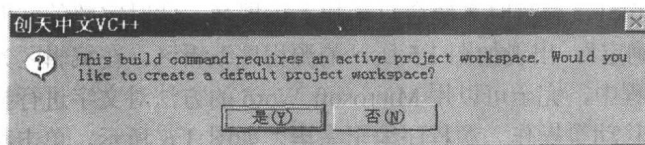


图 1.7 询问是否建立一个默认的工作区

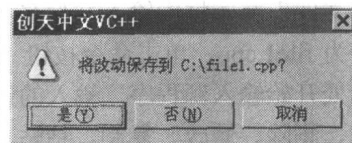


图 1.8 单击命令按钮“是(Y)”

此时在图 1.9 中的输出窗口中将显示:

Compiling...

file1.cpp

file1.obj - 0 error(s), 0 warning(s)

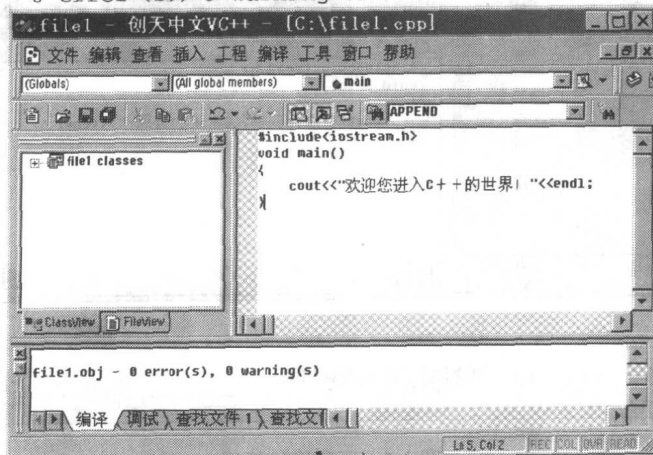


图 1.9 编译成功的输出窗口

它表示编译成功，产生一个目标文件：file1.obj。如果程序有错误，在输出窗口中会指出每一项的错误及产生错误的原因、所在行。

单击菜单“编译”→“执行 file1.exe Ctrl+F5”，如图 1.10 所示，在弹出的图 1.11 中，单击命令按钮“是 (Y)”，将产生一个可执行文件 file1.exe，运行的结果如图 1.12 所示，单击窗口右上角的“关闭”按钮，关闭程序运行结果。

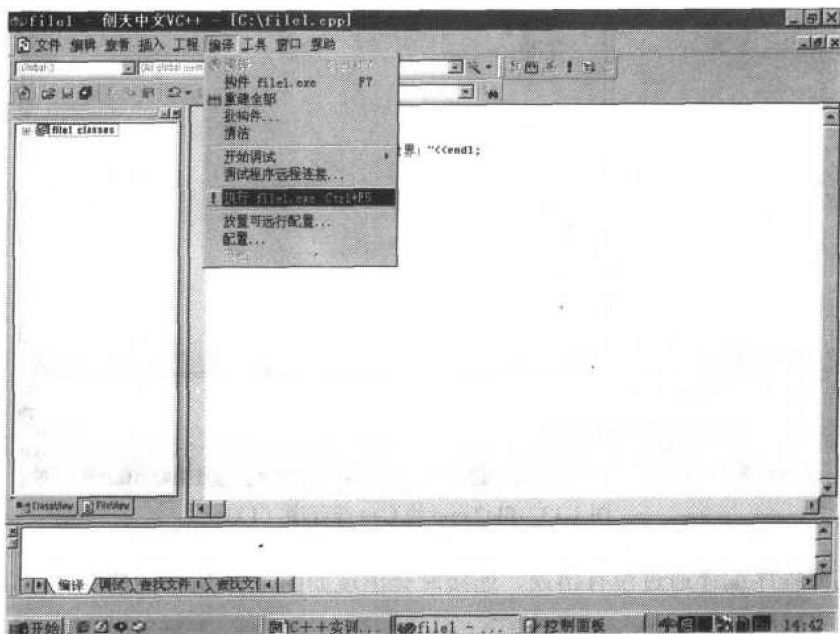


图 1.10 执行已编译好的文件

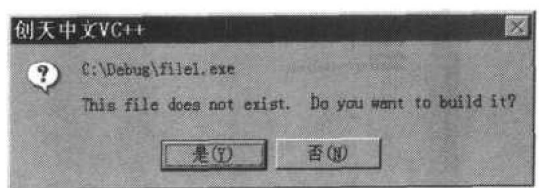


图 1.11 单击命令按钮“是 (Y)”，产生可执行文件 file1.exe

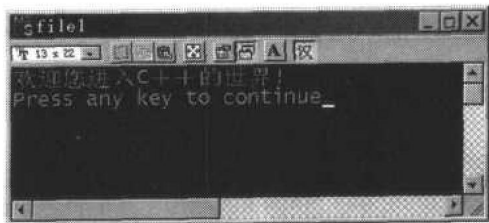


图 1.12 程序运行的结果

不退出 Visual C++再编一个 C++源程序。

不退出 Visual C++再编一个 C++源程序，有时在连接时会出错。单击“文件”→“新

建”，在“新建”对话框中选择“C++ Source File”，在文件框中输入文件名 file2，再编一个名叫 file2.cpp 的 C++源程序，如图 1.13 所示。

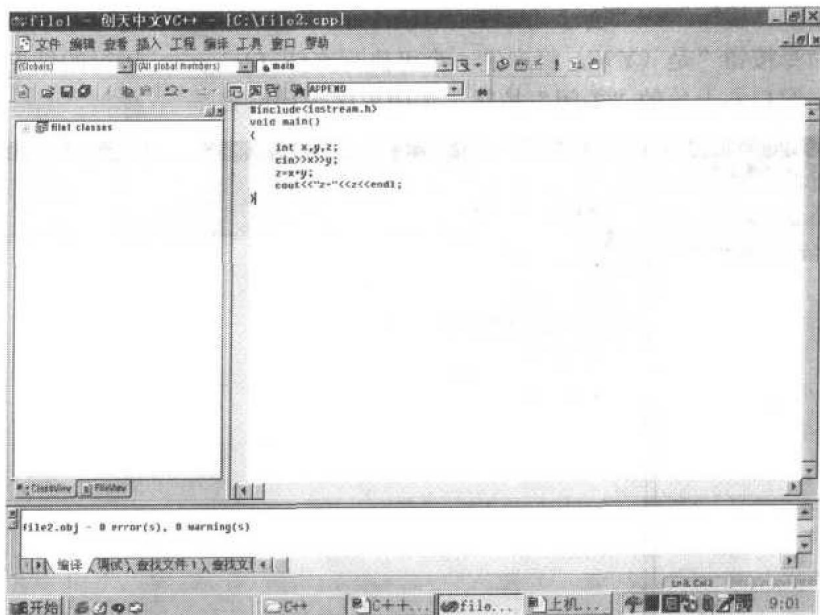


图 1.13 file2.cpp 的 C++源程序 (1)

若该源程序编译通过没有错误，连接时会出现如图 1.14 所示的情况。

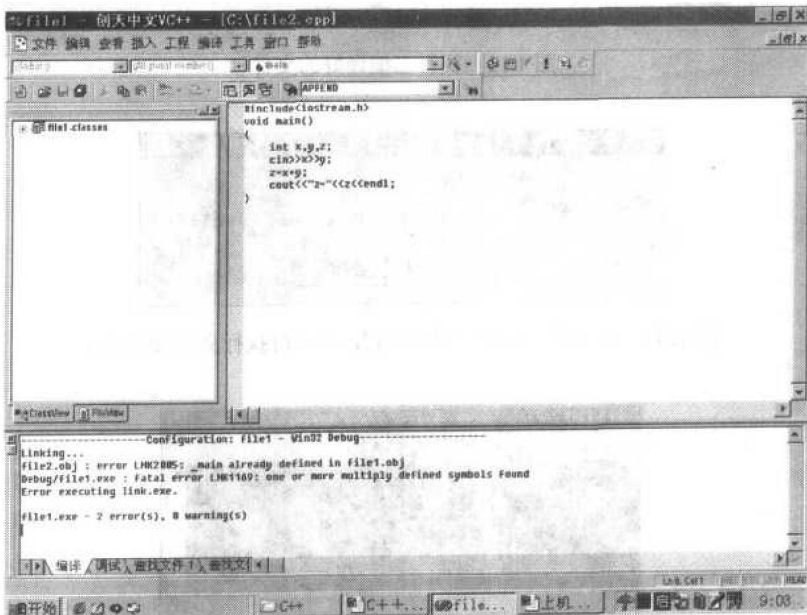


图 1.14 file2.cpp 的 C++源程序 (2)

要解决这个问题，单击菜单的“文件”→“关闭工作区”，关闭前一题的工作空间，重新编译即可。

1.2 Visual C++ 6.0 常用工具栏

Visual C++ 6.0 编程环境由标题栏、菜单栏、工具栏、工作区窗口、代码编辑窗口、输出窗口和状态条组成。屏幕的最上端是标题栏，标题栏下面是菜单栏和工具栏。工具栏下方有两个窗口，左边是工作区或称为工作空间（Workspace），右边的是代码编辑窗口。在工作空间和代码编辑窗口下方是输出窗口（Output），它主要用于显示项目建立过程中生成的各种信息。屏幕的最下端是状态条，它给出当前操作或所选择命令的提示信息。

下面将介绍常用工具：

-  **Open:** 打开已有的文件。
-  **Save:** 保存当前打开的文件。
-  **Save All:** 保存所有打开的文件。
-  **Cut:** 将选定的内容剪切到剪贴板。
-  **Copy:** 将选定的内容复制到剪贴板中。
-  **Paste:** 在光标的当前位置插入剪贴板中的内容。
-  **Workspace:** 显示或隐藏工作区窗口。
-  **Output:** 显示或隐藏输出窗口。
-  **Compile:** 编译文件。
-  **Build:** 建立工程。
-  **Stop Build:** 停止建立工程。
-  **Execute Program:** 执行程序。
-  **Go:** 启动或继续程序的执行。
-  **Insert/Remove Breakpoint:** 插入或删除断点。

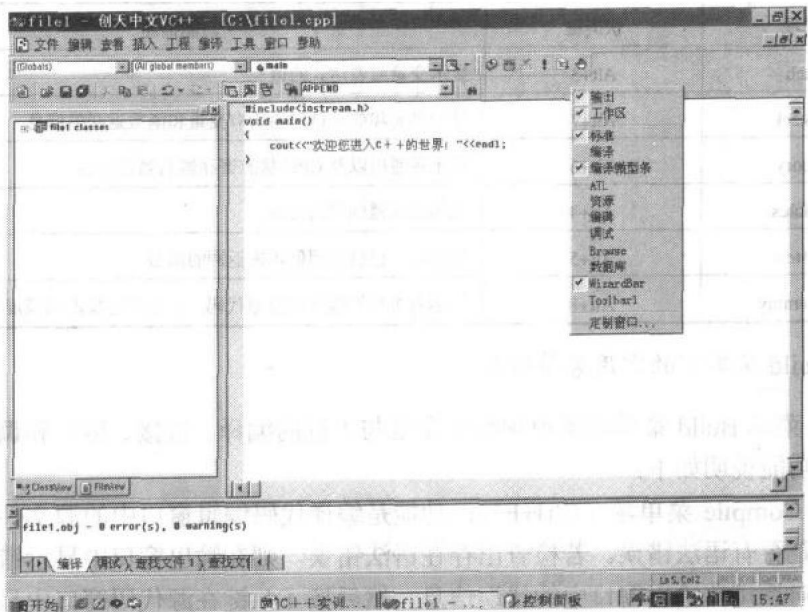


图 1.15 对工具栏的操作

如果用户希望显示或隐藏某个工具栏，可以右键单击工具栏空白处，弹出工具栏快捷菜单，如图 1.15 中左边的菜单所示。工具栏前面带有选中标记“√”，则表示此工具栏已经显示，如图 1.15 中的工作区所示。在此菜单中单击要显示的工具栏，如果要隐藏已经显示的工具栏，可单击相应工具栏清除选中标记。

1.3 Visual C++ 6.0 中常用菜单

在 Visual C++ 开发环境中，大部分的操作都是通过菜单来进行的，因而对菜单命令的基本功能的了解很重要。通过它们，可以轻松地完成编辑程序和资源、编译、链接及调试程序等各项工作。

1. “查看”菜单下的常用菜单命令

(1) 显示工程工作区窗口：单击“查看”→“工作区”，或按 Alt+0 快捷键，功能是显示工程工作区窗口。当工程工作区显示并且是当前活动窗口时，按 Shift+Esc 键则可以将工作区窗口隐藏。

(2) 显示输出窗口：单击“查看”→“输出”，或按 Alt+2 快捷键，功能是显示输出窗口，输出窗口可以显示程序建立过程的有关信息或错误信息，并显示调试运行时的输出结果。当输出窗口显示并且是当前活动窗口时，按 Shift+Esc 键则输出窗口隐藏。

(3) 显示调试信息窗口：单击“查看”→“调试窗口”，当您选中此菜单项时，将弹出子菜单。

表 1.1 Debug Windows 子菜单说明

菜单项	快捷键	功能描述
Watch	Alt+3	显示变量或表达式的值
CallStack	Alt+7	显示当前和前一语句中所有变量和函数返回值信息
Memory	Alt+6	显示各通用以及 CPU 状态寄存器的当前内容
Variables	Alt+4	显示存储器的当前内容
Registers	Alt+5	显示所有已被调用但还未返回的函数
Disassembly	Alt+8	显示有关反汇编代码及原代码，以便进行反汇编或混合调试

2. Build 菜单下的常用菜单命令

Build 菜单 Build 菜单的菜单项命令主要与工程的编译、链接、运行和调试有关，各菜单项的功能说明如下：

(1) Compile 菜单项 (Ctrl+F7)：功能是编译代码编辑窗口中的源文件，以便检查源文件中是否有语法错误。若检查出存在语法错误，则在输出窗口中显示错误信息。可以向前或向后浏览输出窗口中的错误信息，然后按 F4 键在源代码窗口中显示相应的代码行。