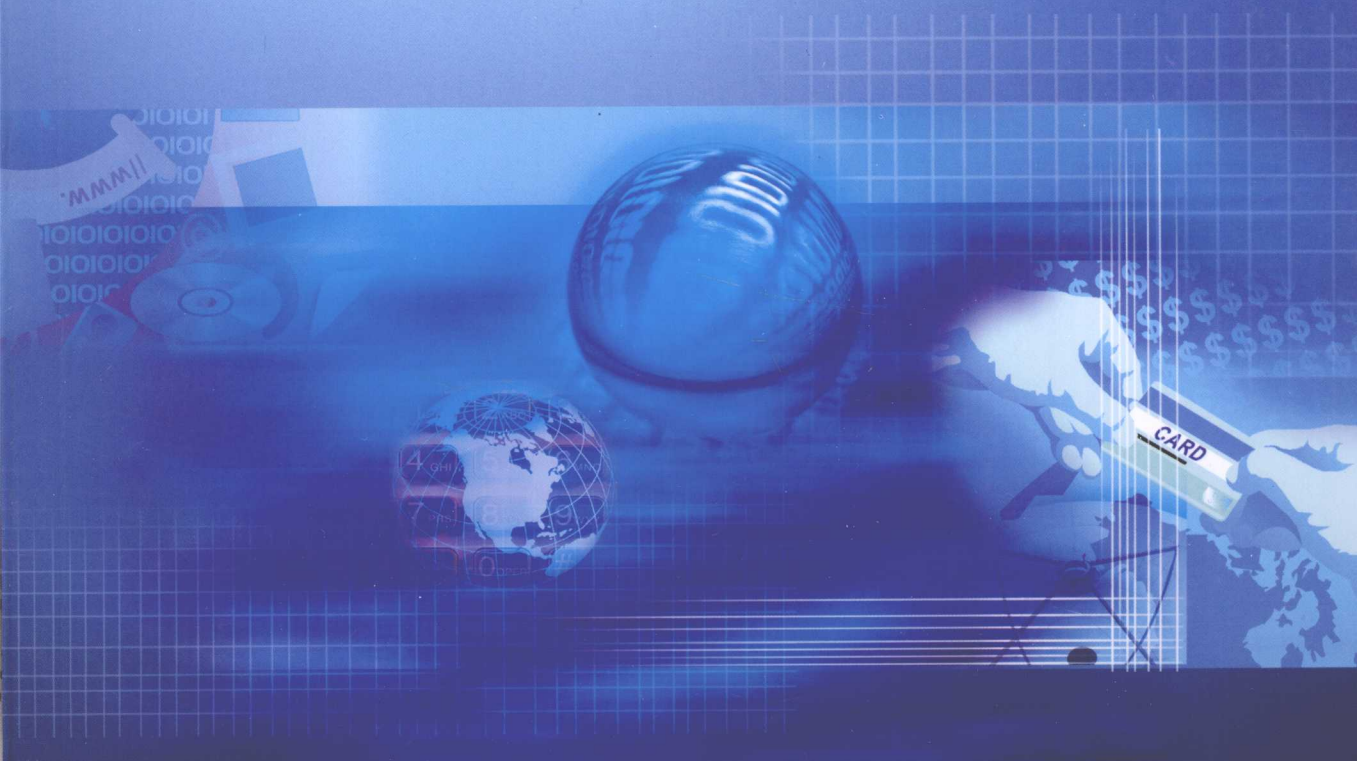




普通高等教育“十一五”规划教材

C语言程序设计实验指导与习题

刘鹏辉 主编 李宏霞 副主编
李银兴 主审



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



普通高等教育“十一五”规划教材

C 语言程序设计实验 指导与习题

刘鹏辉 主 编

李宏霞 副主编

李建卓 参 编

李银兴 主 审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是与《C 语言程序设计》(中国铁道出版社出版刘鹏辉主编)配套的实验指导用书,由实验指导、习题和综合自测题三部分组成。实验指导部分由十个实验组成,介绍了在 Visual C++ 6.0 平台下的 C 程序开发环境,每个实验按调试示例、改错题、编程题顺序分层次、循序渐进地进行实验教学。读者可通过实验掌握基本的程序调试方法,理解和掌握程序设计的思想、方法和技巧。此外,也可通过实验培养读者分析实际问题、编程和动手的能力。习题部分结合教材配有大量的习题,读者可通过习题掌握课本所学知识。综合自测题给出了两套测试题及其答案。

本书以验证和观察型实验为基础,以分析型实验培养读者分析和解决问题的能力,以设计型和综合型实验训练读者设计和编程的能力,掌握 C 语言程序设计的基础知识和综合应用能力。

本书适合作为高等院校《C 语言程序设计》的配套教材和参考用书,也可作为对 C 程序设计感兴趣读者的自学用书。

图书在版编目(CIP)数据

C 语言程序设计实验指导与习题 / 刘鹏辉主编. —北京:
中国铁道出版社, 2009. 11
普通高等教育“十一五”规划教材
ISBN 978-7-113-10561-7

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 194330 号

书 名: C 语言程序设计实验指导与习题
作 者: 刘鹏辉 主编

策划编辑: 严晓舟 滕 云

责任编辑: 杜 鹏

封面设计: 付 巍

版式设计: 于 洋

编辑部电话: (010) 63583215

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 三河市华丰印刷厂

版 次: 2009 年 11 月第 1 版 2009 年 11 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 11.5 字数: 284 千

书 号: ISBN 978-7-113-10561-7/TP·3578

定 价: 21.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

C语言程序设计是一门实践性很强的课程，需要进行大量的实际操作和实践训练。通过该课程的学习，读者不仅可以掌握程序设计语言的知识，更重要的是在实践中逐步掌握程序设计的思想和方法，培养分析、解决问题和应用程序语言的能力。

本书是《C语言程序设计》的配套实验指导用书，介绍了在 Visual C++ 6.0 下的 C 程序开发环境，以验证和观察型实验为基础，以分析型实验培养读者分析和解决问题的能力，以设计型和综合型实验训练读者的设计和编程能力，使读者掌握 C 语言程序设计的基础知识和综合应用能力。

本书由实验指导、习题和综合自测题三部分组成。实验指导部分由十个实验组成，每个实验给出了实验目的和实验内容。实验内容由调试示例、改错题和编程题组成。通过实验使读者掌握基本的程序调试方法，理解和掌握程序设计的思想、方法和技巧，培养读者分析实际问题、编程和动手能力。习题部分结合教材配有大量的习题，使读者通过习题掌握课本所学知识。综合自测题给出了两套测试题及其答案。

本书由刘鹏辉任主编，李宏霞任副主编。实验 1、2、6、7、8、10 由李宏霞编写，实验 3 由刘鹏辉编写，实验 4、5、9 由李建卓编写，习题和综合自测题由李宏霞、刘鹏辉、李建卓编写。李银兴对全书进行了审定，最后由刘鹏辉统稿和定稿。

由于作者水平有限，书中错误和不足之处在所难免，敬请读者批评指正。

编 者

2009 年 8 月

目 录

第一部分 实验指导

实验 1	Visual C++ 6.0 IDE 上机操作过程.....	1
实验 2	输入/输出操作.....	13
实验 3	选择结构.....	24
实验 4	循环结构.....	29
实验 5	数组.....	33
实验 6	函数.....	42
实验 7	变量的定义域与存储方式.....	47
实验 8	指针和命令行参数.....	52
实验 9	结构体和共用体.....	61
实验 10	文件操作.....	68

第二部分 习 题

第 1 章	C 语言概述.....	71
第 2 章	数据类型和运算符.....	74
第 3 章	程序流程控制.....	80
第 4 章	数组.....	97
第 5 章	函数.....	105
第 6 章	指针.....	116
第 7 章	结构体.....	128
第 8 章	编译预处理.....	137
第 9 章	文件.....	142
第 10 章	C 语言图形程序设计.....	147

第三部分 综合自测题

综合自测题（一）.....	150
综合自测题（二）.....	163
综合自测题（一） 参考答案.....	174
综合自测题（二） 参考答案.....	176
参考文献.....	178

第一部分 实验指导

实验 1 | Visual C++ 6.0 IDE 上机操作过程

【实验目的】

1. 学习 Visual C++ 6.0 集成环境，掌握源程序的编辑方法。
2. 掌握在 Visual C++ 6.0 集成开发环境中编译、调试与运行程序的方法。
3. 通过运行简单的 C 程序，初步了解 C/C++源程序的特点。

【实验内容】

一、了解 C/C++程序的开发步骤和上机调试流程

C/C++是编译型语言，设计好一个 C/C++源程序后，需要经过编译、链接生成可执行的程序文件，然后执行。一个 C/C++程序的具体开发步骤如下：

1. 分析问题，产生解题步骤，即解题算法。
2. 根据解题算法编写 C/C++源程序。

3. 利用编辑器编辑源程序并保存，保存文件的扩展名为.cpp。（在 C++开发环境中保存的源程序默认文件扩展名为.cpp）

4. 编译源程序，并产生目标程序。在 Windows 操作系统中，目标程序文件的扩展名为.obj。

5. 连接。将一个或多个目标程序与本程序所引用的库函数进行链接后，产生一个可执行文件。在 Windows 操作系统中，可执行文件的扩展名为.exe。

6. 调试程序。运行可执行文件，分析运行结果。若结果不正确，则要修改源程序，并重复以上过程，直到得到正确的结果为止。

7. 优化。进一步提高程序的运行效率，主要通过改进所用算法，缩短程序运行时间；合理地分配和使用内存，减少程序所用的存储空间。

其中，第 1、2 两步在上机前完成；第 3~6 步在上机时完成，如图 1-1 所示；第 7 步在上机后思考。第 1~6 步是对所有学习编程者的基本要求，第 7 步适用于有兴趣、学有余力且有志成为编程高手者。

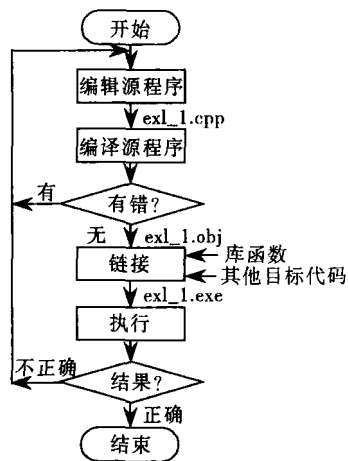


图 1-1 C/C++程序上机调试流程

二、Visual C++ 6.0 编译系统功能简介

1. 窗口。Visual C++ 6.0 的 Developer Studio 编译环境，如图 1-2 所示。



图 1-2 Visual C++ 6.0 的 Developer Studio 编译环境

主窗口由标题栏、菜单栏、工具栏、工作区窗口、程序编辑窗口、输出信息窗口和状态栏组成。

2. 菜单包括“文件”、“编辑”、“查看”、“插入”、“工程”、“组建”、“工具”、“窗口”和“帮助”。

(1) “文件”菜单包括 10 个菜单项。

① “新建”命令。用以打开一个新的编辑窗口。我们通常从这里开始一个新的应用程序。IDE 自动地给打开的每个窗口加上标题和编号。编号从 1 开始，因此第一个窗口的标题通常是 xxx1，第二个窗口的标题是 xxx2，等等。xxx 是区别正在处理文件的类别（如：源程序 Cpp1、Cpp2…，二进制文件 Data1、Data2…，图标 Icon1、Icon2…，等等）的标号。如果我们打开了标题为 xxx1 至 xxx6 的窗口，然后决定关闭标题为加 2 的窗口，则在下一次使用“新建”命令时，这个标题（xxx2）将不会再被使用。Windows 自动提供下一个更大的数字。

打开新建对话框的最快方法是单击工具栏中最左边的按钮。这个按钮上有一个文件的图片，我们可以通过单击这个控件来激活“新建”命令。

② “打开”命令。与“新建”命令不同（这个命令打开一个编辑对话框以编辑以前不存在的文件），“打开”命令打开的对话框需要一些有关以前保存文件的信息。这个对话框叫做“打开文件”对话框，它显示默认驱动器、路径、文件等参数，并允许用户自己设定。

这个对话框有一个节省时间的功能，它可以自动地记住设置并把它作为以后打开时的默认选项。在试图打开一个已经被打开的文件时，会给出声音提示和警告信息，这样可以避免打开同一个文件的两个甚至多个副本，在编辑了其中的一个以后，又保存了没有得到更新的版本。

工具栏上左边第二个按钮上面画着一个文件夹和一个打开箭头，通过这个按钮可以直接激活“打开”菜单。

③ “关闭”命令。用于关闭一个打开的文件。如果打开了多个文件，那么这个命令将关闭活动的或者选中的窗口。从窗口的边框上可以看出哪个窗口是活动的。活动的或者说被选中的窗口中带有键盘和鼠标的输入点（意思是这个窗口接受键盘和鼠标的输入），而且按照系统选定的颜色优先值显示。这些优先值一般包括彩色的标题栏和暗色的窗口边框。不活动的窗口通常具有灰色

的标题栏和边框。

如果在无意中单击了“关闭”按钮，要关闭一个没有保存过的文件，这时也不用担心，集成环境会自动地采取保护措施，它会提示这个文件还没有保存，并且询问现在是否要保存，以此避免造成任何不必要的损失。

④“保存”命令。将当前活动的或被选中的窗口内容保存到指定文件。只要简单地检查一下窗口的标题栏，就可以分辨出窗口中的内容是否已经保存过。如果看到的是一个缺省的标题，如 xxx1，就可以知道这个窗口的内容还从来没有以有效的文件名保存过。在保存一个从来没有保存过的文件时，系统会自动调用“另存为”对话框。

⑤“另存为”命令。可将活动窗口中的内容以新的名字保存一个备份。在什么情况下会用到这个选项呢？可能会遇到这样的情况，刚刚完成一个项目，并有一个正在使用的程序。这时，也许想要试着做一些改动，但又不想破坏现在的版本，出于安全的考虑，可选择“另存为”命令，把文件的内容复制到一个新的文件中，然后就可以放心地更改复制的版本。这时，即使发生了灾难性的错误也不要紧，只要回到原来的文件就行了。

⑥“全部保存”命令。“保存”命令的问题是它只保存活动窗口的内容，“全部保存”命令保存所有窗口的内容，如果有窗口的内容还从来没有被保存过，“全部保存”命令会自动调用“另存为”对话框，要求其输入有效的文件名。

⑦“打开工作区”命令。选择该命令，将弹出“打开工作区”对话框，用来打开该工作区的文件，也可打开其他文件。

⑧“关闭工作区”命令。该命令用来关闭当前工作区的文件。选择该命令后，弹出一个对话框，提示用户是关闭所有文件还是保留这些文件。

⑨“最近文件”命令。该命令用来显示打开过的最近四个文件，单击文件名可以将文件打开。

⑩“退出”命令。该命令用来退出 Visual C++ 6.0 编译系统。在退出系统前，应将打开的文件进行保存。

(2)“编辑”菜单包括 15 个命令。

①“取消”命令。该命令可用以取消最近的编辑操作。我们也可以通过工具栏来实现取消操作。在工具栏上取消是一个向左指向的箭头。

②“重做”命令。该命令可用以恢复最近的取消操作。如果发现曾经认为错误的操作是有效的，就可以通过“重做”命令来重新执行这个操作。“重做”命令也可以通过工具栏来使用。在工具栏中，“重做”命令是一个向右指向的箭头。

③“剪切”命令。该命令将选定的文本从活动窗口中复制到剪贴板，并将它从活动窗口中删除。

④“复制”命令。该命令将选定的文本内容复制到剪贴板上。

⑤“粘贴”命令。该命令用于将剪贴板上的内容插入当前的光标位置。

⑥“删除”命令。该命令删除选定文本而不把它复制到剪贴板上。

⑦“选择全部”命令。该命令选定活动窗口中的所有内容以供剪切、复制或删除。

⑧“查找”命令。该命令的功能是用来在当前打开的文件中查找指定的字符串。

⑨在“文件中查找”命令。该命令在多个文件中搜索指定的字符串。

⑩“替换”命令。“替换”命令激活“替换”对话框，可以用来替换文本。

⑪“定位”命令。通过该命令，我们可以在活动编辑窗口内将光标快速移动到特定位置。

⑫“书签”命令。该命令用以在源程序中最常访问的代码行上加上书签。

⑬ “断点”命令。该命令用以在指定位置设置断点，也可以是选定的数据项或消息。

⑭ “列出成员”命令。该命令显示选定类或结构的有效成员变量或函数的列表。

⑮ “参数信息”命令。该命令显示光标左边函数的完整声明，包括参数列表。粗体参数表示在输入该函数时要求的下一个参数。

(3) “视图”菜单包括五个命令。

① “建立类向导”命令。该命令用来显示 MFC 建立类向导对话框，使用它可以进行 MFC 的常规操作。

② “全屏”命令。该命令用来使源代码编辑区扩大到全屏幕。

③ “工作区”命令。该命令用于显示项目工作区窗口。

④ “输出”命令。该命令用于显示数据输出窗口，在编译时该窗口将会显示出编译及连接信息，包括出错信息。

⑤ “调试窗口”命令。该命令出现级联菜单，在级联菜单中列出了调试窗口的若干操作命令。

(4) “插入”菜单包括三个命令。

① “资源副本”命令。该命令用来复制选定的资源。

② “作为文本文件”命令。该命令用来选择插入到文档中的文件。

③ “新的 ATL 对象”命令。该命令用来启动 ATL 对象向导，将添加新的对象到项目中。

(5) “工程”菜单包括三个命令。

① “设置活动工程”命令。该命令用来选择当前活动项目。

② “插入工程到工作空间”命令。该命令用来将项目插入到工作区中。

③ “增加到工程”命令。该命令用来将新文件或已有文件或部件及控制加到指定的项目中去。

(6) “组建”命令包括 12 个命令。

① “编译”命令。该命令用于编译显示“代码编辑”窗口中的源文件，并检查源文件中的语法是否有错误。

② “组建”命令。该命令用来创建当前文件项目，包含了对源文件或项目的编译和连接，最终生成可执行文件。

③ “全部组建”命令。该命令用来对所有文件进行重新编译、连接，包含已编译过的文件。

④ “运行”命令。该命令用来运行已生成好的可执行文件，并将运行结果显示到相应的环境中。

⑤ “开始调试”命令。选择该命令出现级联菜单，选取该菜单选项便可启动调试器，并用“调试”命令代替“组建”命令。

⑥ Go 命令。该命令用在调试过程中，从当前语句启动或者继续运行。

⑦ Restart 命令。该命令将重新装载程序到内存中，并将放弃所有变量的当前值。

⑧ Stop Debugging 命令。该命令将中断当前调试过程，并返回到原来的编辑状态。

⑨ Step Into 命令。该命令用来设置单步执行程序。

⑩ Step Out 命令。该命令用来在单步执行时，从某个函数体内跳出，调试该函数调用语句后面的语句。该命令与 Step Into 命令配合使用。

⑪ Step Over 命令。该命令也是单步操作命令，只是当程序执行到某一函数调用语句时，不进入该函数体，直接执行该调用语句，然后停在该调用语句后面的语句。

⑫ Quick Watch 命令。选择该命令，将弹出 Quick Watch 对话框，通过该对话框可以查看和修改变量和表达式，或将变量和表达式添加到 Watch 窗口中。

(7) “工具” 菜单。该菜单中的命令用来浏览用户程序中定义的符号、定制菜单与工具栏、激活常用工具或更改选项和变量的设置。

(8) “窗口” 菜单包括六个命令。

- ① “新建窗口” 命令。该命令用来为当前项目打开一个新窗口。
- ② “分割” 命令。该命令用于将活动窗口进行分割。
- ③ “关闭” 命令。该命令用来关闭当前打开的窗口。
- ④ “关闭全部” 命令。该命令用来关闭所有的窗口。
- ⑤ “下一个” 命令。该命令用来显示下一个窗口。
- ⑥ “上一个” 命令。该命令用来显示上一个窗口。

(9) “帮助” 菜单。该菜单中的命令帮助用户获得大量的帮助信息。

3. 项目工作区。项目(工程)是一些相互关联的源文件的集合, 这些源文件组成一个程序, 它们被编译、连接后生成一个可执行文件。

在 Visual C++ 中, 文件、项目和项目配置是由项目工作区组织起来的。项目工作区的内容和设置通过项目工作区文件(.dse)来描述, 在建立一个项目工作区文件的同时, 还生成项目文件(.dsp)和工作区选项文件(.opt), 工作区选项文件用来保存工作区的设置。

项目工作区窗口用来查看和修改项目中的所有元素。该窗口的底部提供了三种面板:

① 类面板(Class View)。该面板在项目工作区窗口中显示该项目中所有类及其成员函数。双击某一项, 则在右边的源代码编辑窗口中显示该成员的源代码。

② 文件面板(file View)。该面板在项目工作区窗口中显示出所有的资源文件及其相互关系。双击某一项, 则在右边的源代码编辑窗口内打开该文件, 显示其源代码。

③ 资源面板(Source View)。该面板在项目工作区窗口中显示项目中的所有资源。双击某一项时, 则在右边的源代码编辑窗口内显示该资源的图形编辑窗口, 可直接在该窗口内增添资源或修改资源特性。

三、Visual/C++ 6.0 上机操作过程

1. C/C++单文件应用程序的实现。

使用 Windows 操作系统的资源管理器, 在 E 盘建立文件夹名为 My (或自己的姓名) 的文件夹。

(1) 新建文件。选择“文件”→“新建”命令, 弹出“新建”对话框, 如图 1-3 所示。

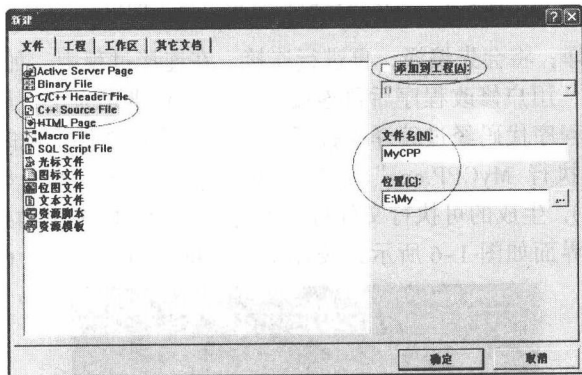











图 1-3 新建 C++源程序

该对话框中有四个选项卡。选择“文件”选项卡，其中列出了 13 个选项。在编辑单个 C/C++ 源文件时，单击 C++ Source File 选项，在右边的“文件名”文本框中输入新建文件的文件名 MyCPP，并在“位置”文本框中指定源文件保存的位置（E:\My\）。单击“确定”按钮返回到 Microsoft Visual C++ 的主窗口。这时可在源代码编辑窗口中输入源代码程序。输入下述源程序：

```
#include<stdio.h>
void main()
{
    printf("Welcom to Visual C++.\n");
}
```

(2) 保存程序。选择“文件”→“保存”（或“另存为”）命令，弹出“保存为”对话框。选择存储文件的路径后，在“文件名”文本框中输入文件名（默认的扩展名为.cpp）后按【Enter】键，或单击“保存”按钮即可。

(3) 编译和连接。     为编译微型条工具栏。

选择“组建”→“编译 MyCPP.cpp”命令（或者单击     编译微型条工具栏中的  Compile 按钮，或者按【Ctrl+F】组合键），将对该源程序代码进行编译。若源程序没有语法错误，则信息窗口显示的信息如图 1-4 所示。

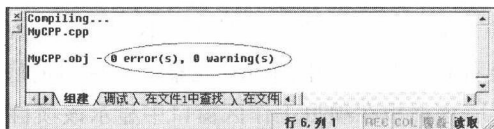







图 1-4 编译正确时信息窗口显示的信息

如果发现错误，则将错误信息显示在下面的输出窗口中，该信息将指出错误性质、出现位置及错误原因等内容。双击某条错误信息时，将会有个提示箭头指向源代码编辑窗口中出错行的右边。用户可以根据错误性质进行修改，然后再重新编译，直到显示没有错误为止。

再选择“组建”→“组建 MyCPP.exe”命令（或者单击     编译微型条工具栏中的  Build 按钮），将对编译好的程序进行连接。连接正确时则信息窗口显示的信息如图 1-5 所示。

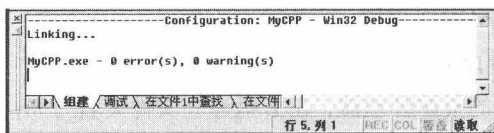







图 1-5 连接正确时信息窗口显示的信息

如果该程序尚未编译，将会先编译，再进行连接。在连接过程中，如果发现错误，仍在输出窗口中显示其错误信息，用户修改程序后再连接，直到没有错误为止，最后生成可执行文件。

(4) 运行程序。源程序代码经过编译连接后，生成可执行文件，此时便可运行。

选择“组建”→“执行 MyCPP.exe”命令（或者单击     编译微型条工具栏中的  BuildExecute 按钮），生成的可执行文件将被运行，运行后的输出结果被显示在 DOS 窗口屏幕上，查看输出结果的界面如图 1-6 所示，按任意键，将返回到 Visual C++ 的主窗口。

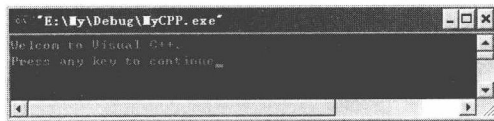


图 1-6 MyCPP.exe 运行界面

2. C/C++多文件应用程序的实现。

(1) 编辑程序中的多个文件。选择“文件”→“新建”命令，弹出“新建”对话框。在该对话框中，选择“文件”选项卡，在该选项卡的列表框中双击 C++ Source 文件选项，弹出 Visual C++ 6.0 主窗口，在源代码编辑窗口中输入其中一个文件 (f1.cpp)，编译没有错误后，选择“文件”→“保存”或“另存为”命令，将文件保存到 E:\MY\1.cpp，选择“文件”→“关闭工作区”命令；再分别输入该程序的其他文件 f2.cpp，编译 f2.cpp 没有错误后，分别将输入的各个文件按各自指定的文件名保存到指定的地方 (E:\MY\2.cpp)，选择“文件”→“关闭工作区”命令。

下面给出一个由两个文件组成的程序的例子。

第一个文件的文件名为 f1.cpp，程序代码如下：

```
#include<stdio.h>
int add(int,int);
void main()
{
    int a,b;
    a=7;b=8;
    printf("%d\n",add(a,b));
}
```

第二个文件的文件名为 f2.cpp，程序代码如下：

```
int add(int x,int y)
{
    return x+y;
}
```

(2) 创建一个空的项目文件，用来存放该程序的上述两个文件。创建一个空项目的方法如下：选择“文件”→“新建”命令，弹出“新建”对话框，选择该对话框中的“工程”选项卡，在该选项卡中，选择 Win32 Console Application 选项，这时在窗口的右下角的“平台”文本框中出现 Win32 复选框，如图 1-7 所示。

在“工程名称”文本框内输入一个项目文件名 MyProject1，然后在“位置”文本框中生成一个路径名 E:\MY\MyProject1，该名可以修改，如图 1-7 所示。

单击“确定”按钮，弹出“Win32 Console Application-步骤 1 共 1 步”对话框，如图 1-8 所示。

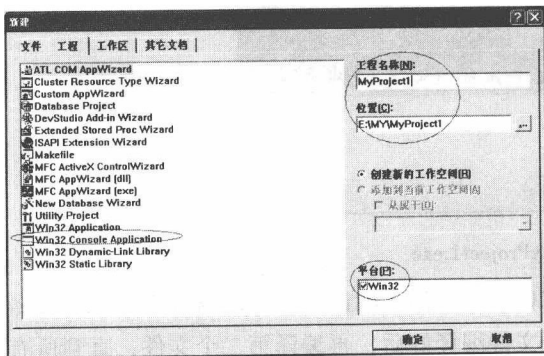


图 1-7 新建工程



图 1-8 新建工程 Win 32 Console Application 步骤 1

该对话框上方有请选择所要创建的控制台应用程序的类型提示信息。这时选择“一个空工程”单选按钮，单击该对话框下方的“完成”按钮。

这时，屏幕上弹出“新建工程信息”对话框，如图 1-9 所示，该对话框告诉用户所创建的控制台应用程序新项目的特性。单击该对话框下方的“确定”按钮，返回到 Visual C++ 6.0 主窗口。项目文件 MyProject1 创建结束。

(3) 将多个文件添加到项目中去。首先，在 Visual C++ 6.0 主窗口中，选择“工程”→“添加到工程”命令，在弹出的级联菜单中选择“文件”命令，弹出“插入文件到工程”对话框，如图 1-10 所示。



图 1-9 “新建工程信息”对话框

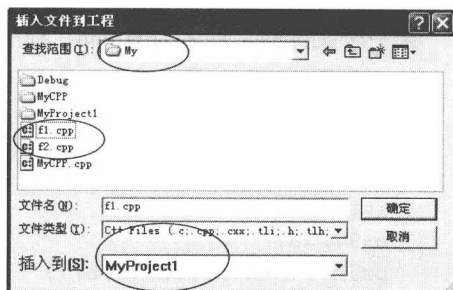


图 1-10 “插入文件到工程”对话框

在该对话框中，先确定项目文件 MyProject1，在“查找范围”下拉列表框中打开 f1.cpp 和 f2.cpp 所在的文件夹，选择这两个文件，然后单击“确定”按钮，则完成添加文件的任务。此时，项目文件 MyProject1 中包含 f1.cpp 和 f2.cpp 文件。

(4) 编译连接项目文件。选择“组建”→“组建 MyProject1.exe”命令，按顺序编译项目中的各文件。未出现错误时，界面如图 1-11 所示。

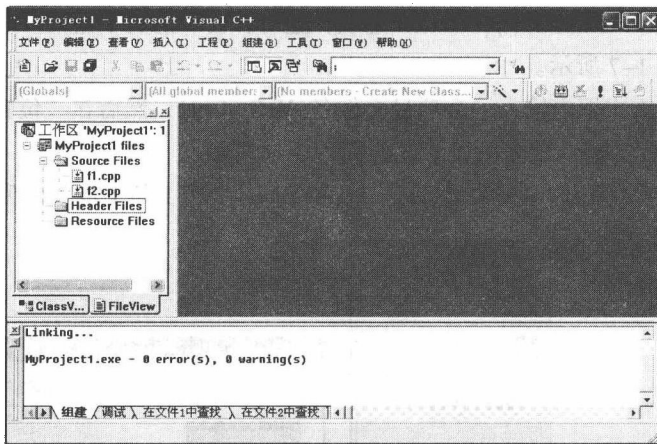



图 1-11 组建 MyProject1.exe

如果发现错误，仍将其错误信息显示在输出窗口中，并停止编译。修改其错误后，继续选择“组建 MyProject1.exe”命令，则重新编译，第一个文件编译好后，再编译第二个文件，直到所有文件都编译好后，再进行连接，直到连接无错。

(5) 运行项目文件。选择“组建”→“执行 MyProject1.exe”命令（或者单击  编

译微型条工具栏中的  BuildExecute 按钮) 生成可执行文件 MyProject1.exe, 并执行该项目文件, 将输出结果显示在 DOS 窗口中。

(6) 关闭当前的项目文件的方法。选择“文件”→“关闭工作区”命令。

注意: 在调试一个应用程序时, Visual C++集成环境一次只能打开一个项目文件。当一个程序调试完成后, 要开始输入另一个程序时, 必须先关闭当前的项目文件, 然后为新源程序建立一个新的项目文件, 否则将出现不可预测的错误。

(7) 退出 Visual C++集成环境。选择“文件”→“退出”命令, 可以退出 VC++ 6.0 的集成环境。

(8) 重新进入 VC++ IDE, 试着调出刚存入的程序文件。再次调出 MyProject1 工作区文件的方法有以下三种:

- 使用资源管理器, 在 E:\My\MyProject1\目录下找到 MyProject1.dsw(此为 MyCpp 工作区文件), 双击即可。
- 先启动 VC++, 再选择“文件”→“最近工作空间”→“E:\My\MyProject1\ MyProject1.dsw”命令即可。
- 先启动 VC++, 再选择“文件”→“打开工作空间”命令, 弹出“打开工作空间”对话框(与文件对话框类似), 在 E:\My\MyProject1 目录下找到 MyProject1.dsw, 单击“打开”按钮即可。

(9) 用资源管理器查找 MyProject1 工作区的下列有关文件并观察其文件大小:

- ① 在 E:\My\ Project1 目录下查找工作区、工程和源程序文件。
- ② 在 E:\My\MyCpp\Debug 目录下查找目标文件和可执行文件。

四. 调试示例

改正下列程序中的错误, 在屏幕上显示内容为 Welcome to You 。

源程序 error01_1.cpp (有错误的程序):

```
int Main(void)
{
    print(Welcome to You\n")
    return 0;
}
```

1. 按照本实验(实验 1)的“三、Visual C++ 6.0 上机操作过程”中的“1.C/C++单文件应用程序的实现”, 将下面的参考程序进行编辑, 保存到 E:\MY\error01_1.cpp 中。

2. 选择“文件”→“打开”命令, 在弹出的对话框中选择要打开的 E:\MY\error01_1.cpp 文件。

3. 编译。选择“组建”→“编译 error01_1.cpp”命令, 信息窗口显示编译错误信息, 如图 1-12 所示。

4. 找出错误。在信息窗口中双击第一条错误信息, 编辑窗口就出现一个箭头指示程序出错的位置, 如图 1-13 所示。一般情况下, 在箭头的当前行或上一行, 可以找到出错的语句。图 1-13 所示为箭头指向第 3 行, 错误信息指出 print 是一个未定义的标识符, 出错原因在于函数名称写错了, 将 print 改为 printf 即可。

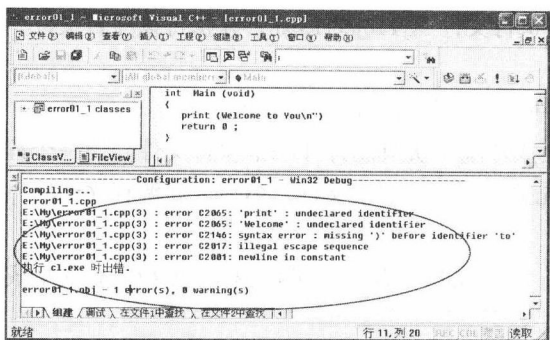


图 1-12 编译产生的错误信息 1

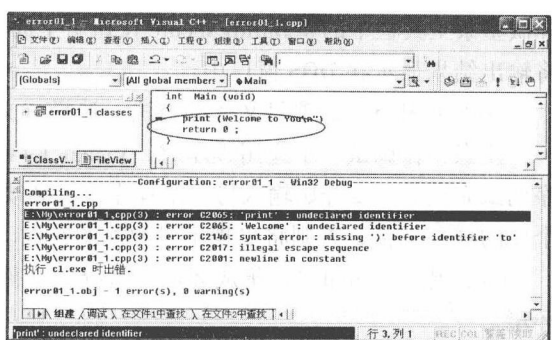


图 1-13 编译产生的错误信息 2

5. 重新编译。信息窗口显示本次编译的错误信息，如图 1-14 所示。双击第一行错误信息，箭头指向出错位置，错误信息指出 printf 是一个未定义的标识符。这次出错原因在于文件没有加预处理命令 #include <stdio.h>。

6. 改错误。在原文件中第一行添加该命令行 #include <stdio.h>。

7. 重新编译。信息窗口显示本次编译的错误信息，如图 1-15 所示。双击第一行错误信息，箭头指向出错位置，错误信息指出 Welcome 是一个未定义的标识符，但 Welcome 并不是标识符。这次出错原因在于 Welcome 前少了前双引号。

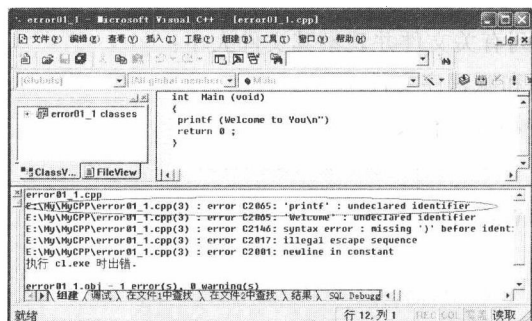


图 1-14 编译产生的错误信息 3



图 1-15 编译产生的错误信息 4

8. 改正错误。在 Welcome 前加上前双引号。

9. 重新编译。信息窗口显示本次编译的错误信息，如图 1-16 所示。双击第一行错误信息，箭头指向出错位置，错误信息指出 return 前少了分号“;”，改正错误，在 return 前加上一个分号。

10. 再次编译。信息窗口显示编译正确，如图 1-17 所示。

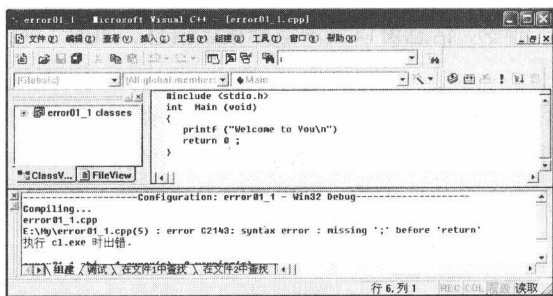


图 1-16 编译产生的错误信息 5

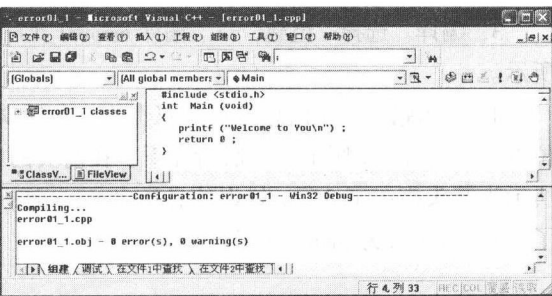


图 1-17 信息窗口显示编译正确

11. 连接。选择“组建”→“组建 error01_1.exe”命令，信息窗口显示连接错误信息，如图 1-18 所示。仔细观察后发现，主函数名 main 首字母被大写，被误写为 Main。

12. 改正错误。把 Main 改写为 main 后，重新编译和连接，信息窗口中没有出现错误信息，如图 1-19 所示。

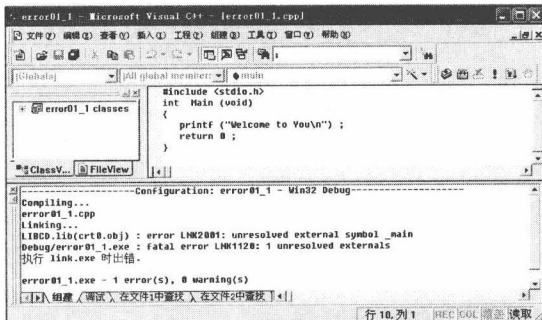


图 1-18 信息窗口显示连接错误信息

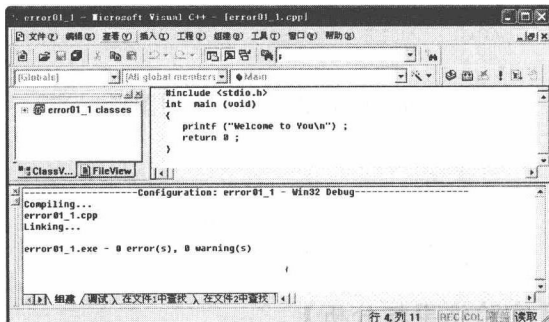


图 1-19 编译正确

13. 运行。选择“组建”→“执行 error01_1.exe”命令，自动弹出运行窗口，如图 1-20 所示。

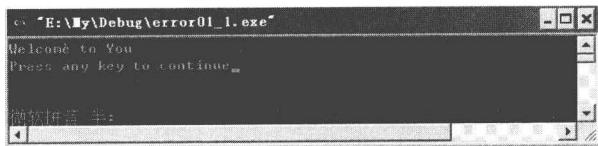


图 1-20 程序运行窗口

五、编程题

1. 设计一个 C++ 程序，在屏幕上显示以下信息：

```
*****
Hello!
*****
```

思考：

- (1) 如何在屏幕上显示自己的学号、姓名和班级？
- (2) 如何在屏幕上显示数字、英文和汉字信息？例如“我学过 Word 2007。”。

参照本实验（实验一）的实验内容三：VC++ 6.0 上机操作过程中的“1. C/C++ 单文件应用程序的实现”，将下面的参考程序进行编辑、编译、连接和运行。

参考程序：

```
#include<stdio.h>
void main()
{
    printf("*****\n");
    printf("    Hello!    \n");
    printf("***** \n");
}

```

运行结果：

```
*****
```


Hello!

2. 在屏幕上显示下列图形。

```
* * * * *
* * * *
* * *
* *
*
```

思考：如何在屏幕上显示下列图形。

```
 *
* * *
* * * * *
* * *
 *
```

【实验结果与分析】

将源程序、运行结果和分析以及实验中遇到的问题和解决问题的方法，写在实验报告上。