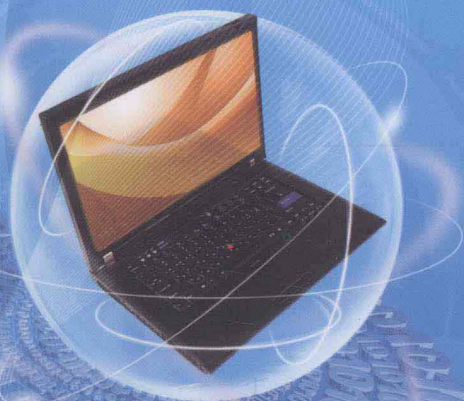




全国高等教育“十二五”精品教材

C++ 程序设计实验 与习题指导

主编 田秀霞 徐建平 彭源



航空工业出版社

全国高等教育“十二五”精品教材

C++程序设计实验与 习题指导

主 编 田秀霞 徐建平 彭 源

副主编 张安勤 温 蜜 毕忠勤

楼越焕 陈宗民

航空工业出版社

北 京

内 容 提 要

本书是《C++程序设计教程》配套的学生用书,书中包含了配套教材中对应各个章节的学习导读、实验指导和课后习题答案,以及为了帮助初学者更好地掌握所学内容而补充的习题。其中,学习导读部分指出了如何学习本章内容,应该重点理解和掌握什么知识;实验指导部分包括了实验目的、实验内容、实验参考三部分内容,它通过精心设计的实验,让读者很好地将理论与实践相结合,从而深刻理解和掌握本章所学内容;习题参考答案部分给出了对应教材中各章习题的参考答案,并对部分选择题给出了简单说明。

图书在版编目(CIP)数据

C++程序设计实验与习题指导 / 田秀霞, 徐建平, 彭源主编. — 北京: 航空工业出版社, 2011.8

ISBN 978-7-80243-813-2

I. ①C… II. ①田… ②徐… ③彭… III. ①C语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第168141号

C++程序设计实验与习题指导 C++ Chengxu Sheji Shiyan yu Xiti Zhidao

航空工业出版社出版发行

(北京市安定门外小关东里14号 100029)

发行部电话: 010-64815615 010-64978486

北京市科星印刷有限责任公司印刷

全国各地新华书店经售

2011年8月第1版

2011年8月第1次印刷

开本: 787×1092

1/16

印张: 18.5

字数: 462千字

印数: 1—3000

定价: 35.00元

编 者 的 话



C++程序设计是一门实践性很强的课程，深刻理解并掌握 C++编程需要重视并增强实践环节，而在实践中遇到的难题和相应的解决方法是仅仅通过听课和阅读教材无法获得的，需要在解决问题的过程中不断积累，通过对一个个问题的解决，如从简单的遗漏分号问题到解决逻辑实现问题，读者会自然而然加深对 C++编程的理解和掌握。

在教学过程中，学生经常会出现这样的问题：听老师讲课，好像什么都明白，但一旦自己动手编程，却又茫然不知所措。例如，初学者经常会问：为什么 C++程序中一定要引入头文件（`#include<iostream>`），为什么一定要包含标准命名空间（`using namespace std;`），以及为什么一定要包含 `main` 主函数，且 `main` 主函数只能有一个，程序才能运行。又如，上课一再强调的容易出错的问题，学生还是不能够完全理解，从而导致上机时出现大量的错误。实际上，这些都是学生，特别是初学者或自学者，在学习过程中必然经历的一个阶段。

我们认为，要学好程序设计，有这么几个关键点：（1）每种编程语言都有自己的各种规定，大家必须了解这些规定并按规定格式书写程序，编译系统才能“理解”和“翻译”它；（2）大家必须完全理解程序中每条语句的意义，即不仅要知其然，还要知其所以然；（3）大家除了要认真学习语言的各种规定外，必须多实践，这也是最重要的一点。只有多实践，大家才能熟练掌握语言的各种规定，才能知道如何将各种问题的解决方法“转化”为程序，才能掌握各种编程技巧，并树立严密的逻辑思维。

我们在 2010 年 9 月份出版了《C++程序设计教程》一书，本书则是该书的配套用书，目的在于进一步为读者学习不同章节知识提供指导，帮助读者提高编程实践能力，使读者在实践过程中有重点地学习，并减少学习中的困惑，从而进一步提高读者深入学习 C++的兴趣。本教材中的实验和相关指导在实际教学中试用多次，并取得了良好的教学效果。

在本书中，每章内容分为 3 部分：第一部分是“学习导读”，主要介绍了学习各章内容时需要重点理解并掌握的要点以及建议的学习方法；第二部分是对应章节的“实验指导”，包括实验目的、实验内容和参考程序 3 个部分。首先，教师可以根据给定的实验题目并结合自己学生的特点灵活选择实验内容；其次，参考程序则给学生进行规范化编程提供了样例说明和指导；第三部分详细给出了《C++程序设计教程》各章课后习题的参考答案。必须说明的是，我们这里只是给出了多种参考答案中的一种，而且不一定是最佳答案，读者可以在答案的基础上继续扩充完善。

此外，为了帮助读者进一步巩固所学内容并提高编程实践能力，我们还为每个章节补充了不同类型的习题供读者练习。

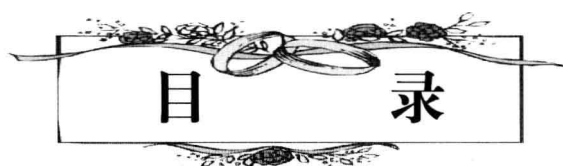
最后要说明的是，本书中的全部程序都在 Windows 环境下的 Visual C++6.0 集成开发环境中测试通过。

感谢您选择使用本书，欢迎您对本书内容提出宝贵的意见和建议。作者的邮件地址是：

c_plusplus@126.com

编 者

2011 年 7 月



目 录

第1章 C++及其开发环境概述	1
一、实验目的	1
二、实验内容	1
三、参考程序	8
课后习题解答	11
第2章 C++程序设计基础（上）	13
一、实验目的	14
二、实验内容	14
三、参考程序	18
课后习题解答	18
第3章 C++程序设计基础（下）	24
一、实验目的	25
二、实验内容	25
三、参考程序	26
课后习题解答	28
第4章 函数的应用	38
一、实验目的	39
二、实验内容	39
三、参考程序	40
课后习题解答	41
第5章 数组与字符串	54
一、实验目的	55
二、实验内容	55
三、参考程序	55
课后习题解答	58
第6章 指针与引用	70
一、实验目的	71
二、实验内容	71
三、参考程序	72
课后习题解答	73
第7章 自定义数据类型	81
一、实验目的	82
二、实验内容	82



三、参考程序	83
课后习题解答	85
第8章 类与对象	113
一、实验目的	115
二、实验内容	115
三、参考程序	116
课后习题解答	118
第9章 继承与派生	130
一、实验目的	131
二、实验内容	132
三、参考程序	132
课后习题解答	138
第10章 多态性、虚函数与运算符重载	150
一、实验目的	151
二、实验内容	151
三、参考程序	152
课后习题解答	153
第11章 输入/输出流	168
一、实验目的	169
二、实验内容	169
三、参考程序	169
课后习题解答	172
第12章 命名空间与异常处理	184
一、实验目的	185
二、实验内容	185
三、参考程序	186
课后习题解答	186
第13章 模板	193
一、实验目的	193
二、实验内容	193
三、参考程序	194
课后习题解答	194
补充习题汇集	203
第2章补充习题	203
第3章补充习题	208
第4章补充习题	220
第5章补充习题	228
第6章补充习题	234
第7章补充习题	237

第 8 章补充习题	244
第 9 章补充习题	257
第 10 章补充习题	275
第 11 章补充习题	279
第 12 章补充习题	283
第 13 章补充习题	284

第 1 章 C++ 及其开发环境概述

【学习导读】

本章是学习 C++ 的入门篇章，更是进入 C++ 编程世界的基础篇章，目的是帮助读者了解面向对象编程和面向结构编程的基本特点，通过对比两种编程方法初步体会面向对象编程的优势所在。本章通过一个简单程序的引入使得读者初步理解 C++ 的基本编程流程，以及 C++ 开发环境的基本使用方法，从而为后继章节的学习奠定基础。

为什么在开始章节就引入两种不同的编程方法呢？一方面，是为了让读者了解面向对象编程和面向过程编程的不同，如编程风格、语法等，进而通过实例程序使读者初步体会 C++ 面向对象编程比面向过程编程带来的优势，如面向对象的思想与人的思维基本一致，而面向过程的编程则主要侧重于算法实现。

另外，1.2 节简单介绍了 C++ 的历史演变过程，读者只需要简单了解；1.3 节介绍了 C++ 程序的开发过程和 C++ 程序的结构和书写形式。在这部分接触的术语不需要死记硬背，大家只要根据给定的实例熟悉 C++ 简单程序的开发流程，以及如何运行自己的第一个 C++ 程序就可以了。

本章的实验部分主要介绍了 C++ 集成开发环境 VC++ 6.0 的简单使用方法，例如，运行 C++ 程序的方法，各个菜单中常用命令的功能及用法，特别是单步执行命令，此命令非常有用，读者可通过单步执行程序理解一些基本程序结构，如选择、循环、函数等的执行过程。

一、实验目的

- 了解 VC++ 6.0 的特点并熟悉 VC++ 6.0 集成开发环境。
- 学习用 VC++ 6.0 编写标准的 C++ 控制台程序，掌握 C++ 控制台程序的开发调试步骤。
- 通过编写、运行简单程序，理解程序的基本结构。
- 学会使用 VC++ 6.0 开发环境中的 debug 调试功能：单步执行、设置断点、观察变量值。

二、实验内容

（一）启动 VC++ 6.0

操作系统：Windows 2000 Server/XP Professional

集成开发环境：VC++ 6.0



微机：486 或以上

(1) 单击“开始”按钮，选择“开始”菜单的“程序”，如图 1-1 所示。

(2) 选择“程序”中的“Microsoft Visual Studio 6.0”，再选择其下的“Microsoft Visual C++ 6.0”，单击后进入 VC++ 6.0 的集成开发环境，如图 1-2 所示。



图 1-1 “开始”菜单

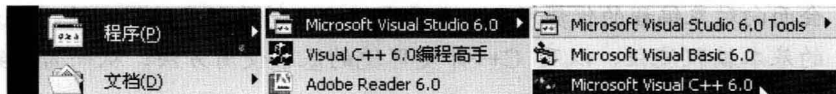


图 1-2 VC++ 6.0 的集成开发环境

(二) 编写并运行第一个标准 C++ 程序

1. 创建新工程

(1) 选择 File 菜单中的 New 选项，打开 New 对话框，如图 1-3 所示。

(2) 单击 Projects 标签，在其中单击 Win32 Console Application (Win32 标准控制台程序)。

(3) 在 Project name 编辑框中输入一个工程名，如 experiment，如图 1-4 所示。

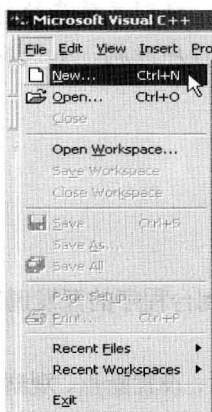


图 1-3 New 菜单

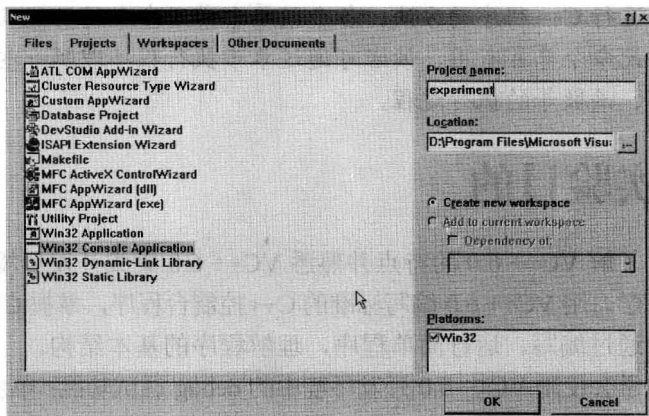


图 1-4 New 对话框中的 Projects 标签

(4) 在 Location 编辑框中输入工程存放路径，也可单击 Location 编辑框右侧的 按钮，打开 Choose Directory 对话框，用鼠标选择工程存放路径，如图 1-5 所示。

(5) 单击 OK 按钮，打开图 1-6 所示 Win32 Console Application (Win32 控制台应用程序) 对话框。

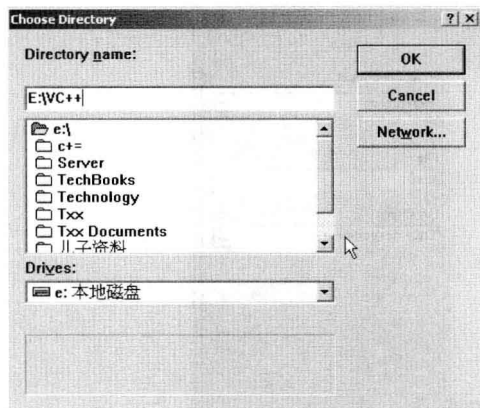


图 1-5 Choose Directory 对话框

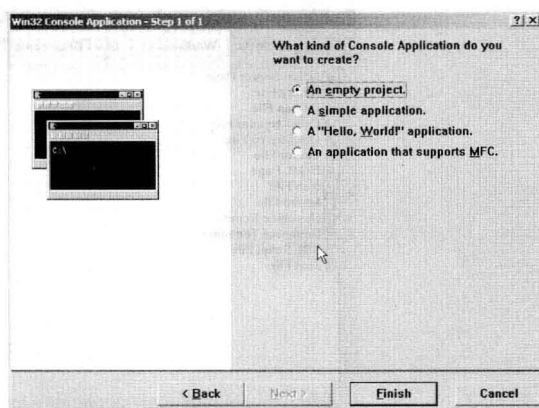


图 1-6 Win32 Console Application 对话框

(6) 选择 An empty project 单选项, 单击 Finish 按钮, 打开图 1-7 所示 New Project Information (新工程信息) 对话框。

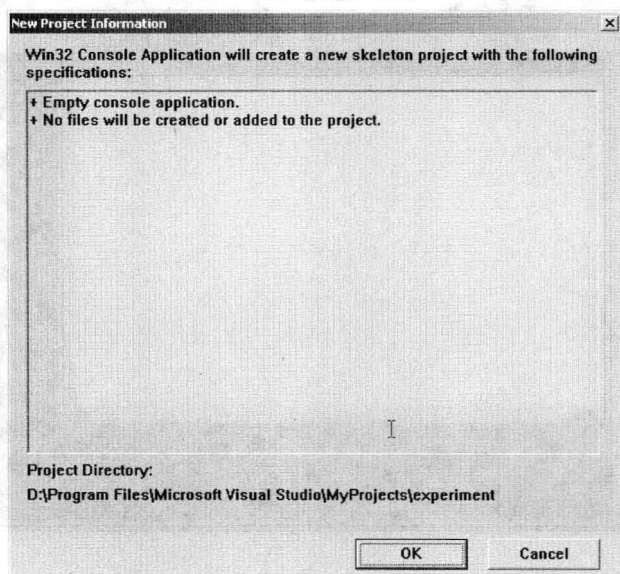


图 1-7 Win32 控制台应用程序窗口

(7) 单击 OK 按钮, 完成工程创建。

2. 建立 C++源程序文件

(1) 选择 File 菜单中的 New 选项, 打开 New 对话框, 如图 1-8 所示。

(2) 单击 Files 标签, 在其中单击 C++ Source File (C++源程序文件)。

(3) 选中 Add to project 复选框, 框中工程名为我们刚刚新建的工程 experiment。

(4) 在 File 编辑框中输入一个文件名(用户自己定义的 C++源程序名), 如 first; Location 编辑框中显示了默认的 C++源程序文件的存放路径。

(5) 单击 OK 按钮, 完成新建 C++源程序文件操作, 打开图 1-9 所示源程序编辑窗口。

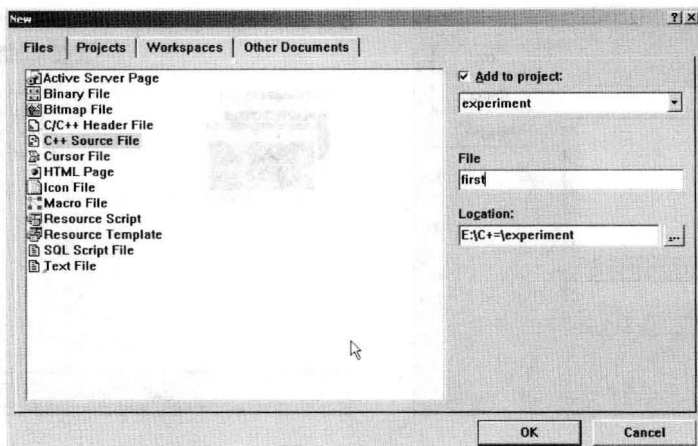


图 1-8 New 对话框

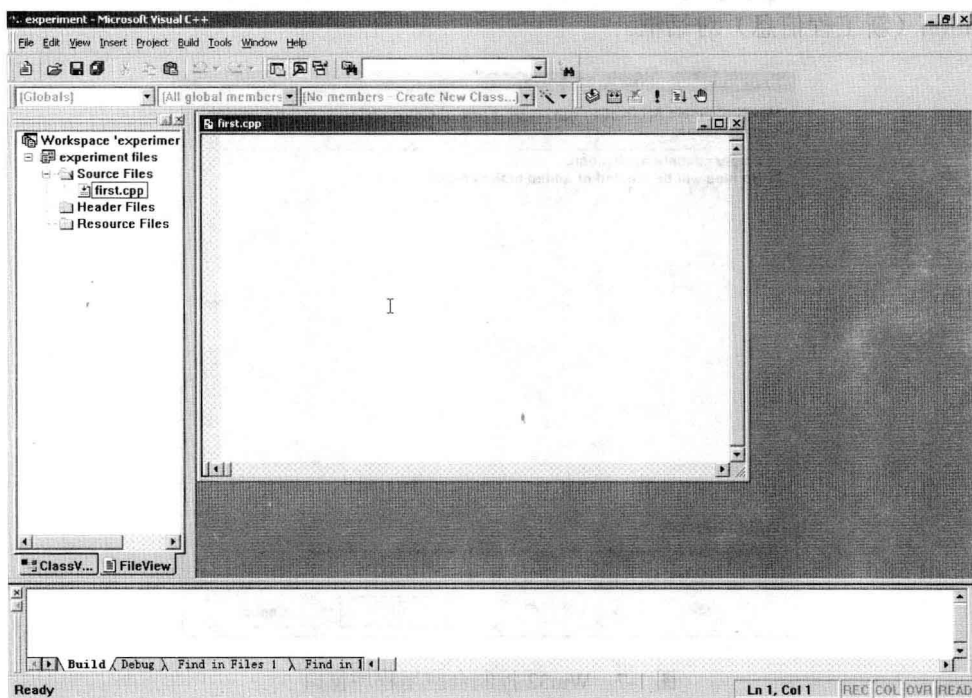


图 1-9 源程序编辑窗口

3. 编辑 C++源程序文件

- (1) 在源程序编辑窗口中输入程序源代码，如图 1-10 所示。
- (2) 选择 File 菜单中的 Save 选项，保存文件。

4. 建立并运行可执行程序文件

- (1) 单击图 1-11 中的编译工具按钮，将该源程序编译为 C++目标代码。如果没有错误，则可以进入下一步链接程序，否则需要修改源程序，再编译，直到没有错误。

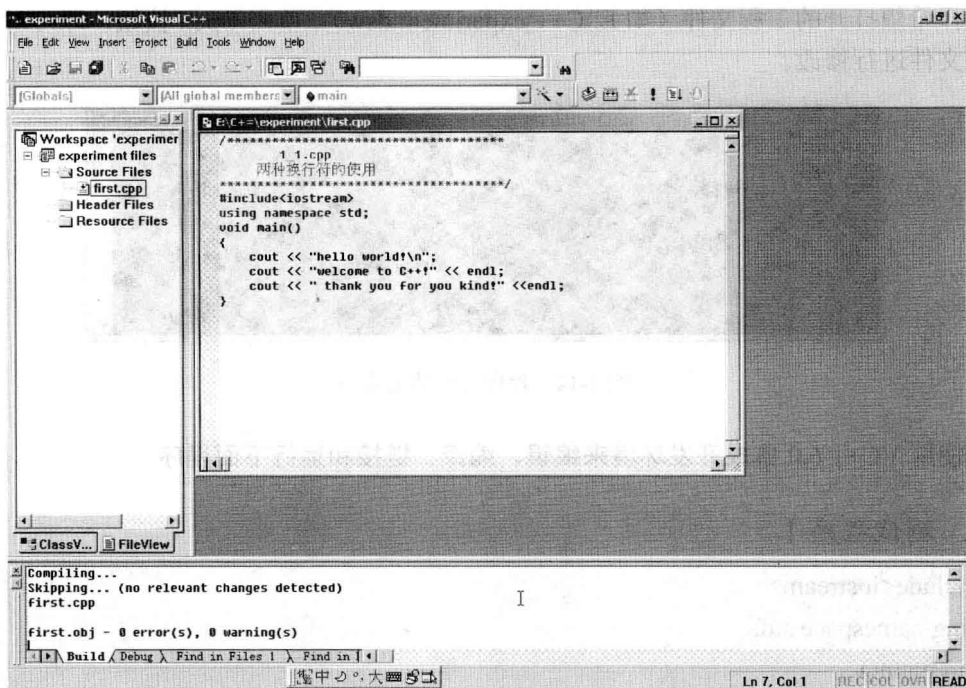


图 1-10 输入源程序

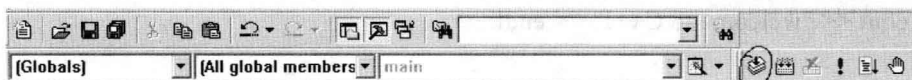


图 1-11 编译按钮

(2) 单击图 1-12 中的链接工具按钮，链接 C++ 目标代码和其他相关库资源。如果没有错误，可以进入下一步运行程序，否则需要查看链接资源是否正确或存在。

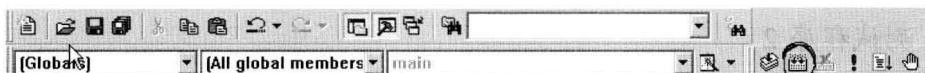


图 1-12 链接按钮

(3) 单击图 1-13 中的执行工具按钮，运行程序，弹出程序运行结果窗口，如图 1-14 所示。如果程序要求输入，还需要通过屏幕输入相应的数据。

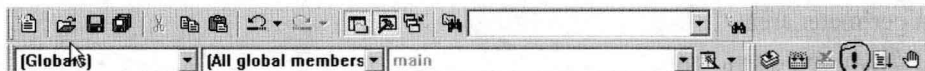


图 1-13 运行按钮

5. 关闭/打开工作区

(1) 选择 File 菜单中的 Close Workspace 选项，可关闭工作区

(2) 要打开工作区，可选择 File 菜单中的 Open Workspace 选项，在打开的 Open 对话



框中选择希望打开的工程文件（如 E:\C++\experiment.dsw），单击 Open 按钮，即可对已建立的工程文件进行修改。

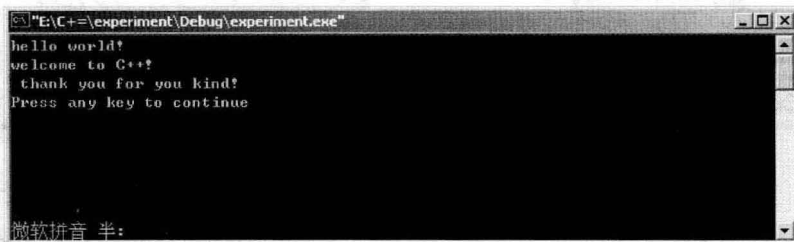


图 1-14 程序运行结果窗口

（三）使用 VC++ 6.0 集成开发环境来编辑、编译、链接和运行下面程序

1. 测试程序 1

```
#include<iostream>
using namespace std;
void main()
{
    cout << "hello world!\n";
    cout << "welcome to C++!" << endl;
}
```

上机要求：

- ① 去掉程序中代码 using namespace std; 观察是否影响程序的执行？
- ② 了解\n 和 endl 意义，观察其作用是否相同？
- ③ 主函数能不能缺省？它的类型能否改变？改变后仔细观察编译时所提示的问题。

2. 测试程序 2

```
#include <iostream>
using namespace std;
void main()
{
    int perimeter,area;
    int length=20,width=10;
    perimeter=2*(length+width);           //求周长公式
    area= length*width;                   //求面积公式
    cout<<"perimeter="<<perimeter<<endl; //输出结果
    cout<<"area="<<area<<endl;
}
```

上机要求:

- ① 将 void 换成 int, 重新编译程序, 观察是否有警告 (warning) 出现? 什么警告?
- ② 将 #include <iostream> 与 using namespace std; 换成 #include <iostream.h>, 重新编译程序, 观察程序运行情况。
- ③ 单步执行测试程序 2, 熟练掌握执行程序各快捷键的用法。

3. 测试程序 3

```
#include <iostream>
using namespace std;
class Rectangle                                //定义长方形类
{
public:
    Rectangle(float w=0,float l=0)            //定义构造函数, 完成创建对象时的
    {
        width=w;length=l;                    //初始化工作, 即为长和宽赋初值
    }
    void GetArea()                             //定义求面积方法
    {
        cout<<"area="<<width*length<<endl;
    }
    void GetPerim()                           //定义求周长方法
    {
        cout<<"perimeter="<<2*(width+length)<<endl;
    }
private:
    float width,length;                       //外界不可访问的私有成员
};
void main()
{
    Rectangle a(10,20);                       //定义长方形类的一个变量 a, 即实例化一个
                                              //特殊的长方形 a, 它的长是 20, 宽是 10
    a.GetPerim();                             //调用 a 对象的两个方法
    a.GetArea();
}
```

上机要求:

- ① 观察测试程序 3 和测试程序 2 的运行结果。
- ② 单步执行测试程序 3。
- ③ 根据程序步骤, 简单理解面向对象程序设计的基本方法。



(四) 运行下面程序, 改正错误

```
#include <iostream>
using namespace std;
int main()
{
    int a=4;
    cout>>"Enter a number: \n";
    b=a*12;
    cout>>"b=\n">>b>>endl;
}
```

上机要求:

- ① 编译程序, 写出编译错误原因。
- ② 根据前面的正确程序修改程序, 并重新编译。

三、参考程序

```
#include <iostream>
using namespace std;
void main()
{
    int perimeter,area;
    int length=20,width=10;
    perimeter=2*(length+width);           //求周长公式
    area= length*width;                   //求面积公式
    cout<<"perimeter="<<perimeter<<endl;   //输出结果
    cout<<"area="<<area<<endl;
}
```

上机要求:

① 将 void 换成 int, 重新编译程序, 观察是否有警告 (warning) 出现? 如有警告, 是什么警告?

答: warning C4508: 'main': function should return a value; 'void' return type assumed

② 将#include <iostream>和 using namespace std; 换成 #include <iostream.h>, 重新编译程序, 观察程序运行情况。

答: 没区别

③ 单步执行测试程序 2, 熟练掌握执行程序的各快捷键的用法。

答:

(a) 选择 Build>Stark Debug>Step Into 菜单 (参见图 1-15), 打开图 1-16 所示调试窗口。

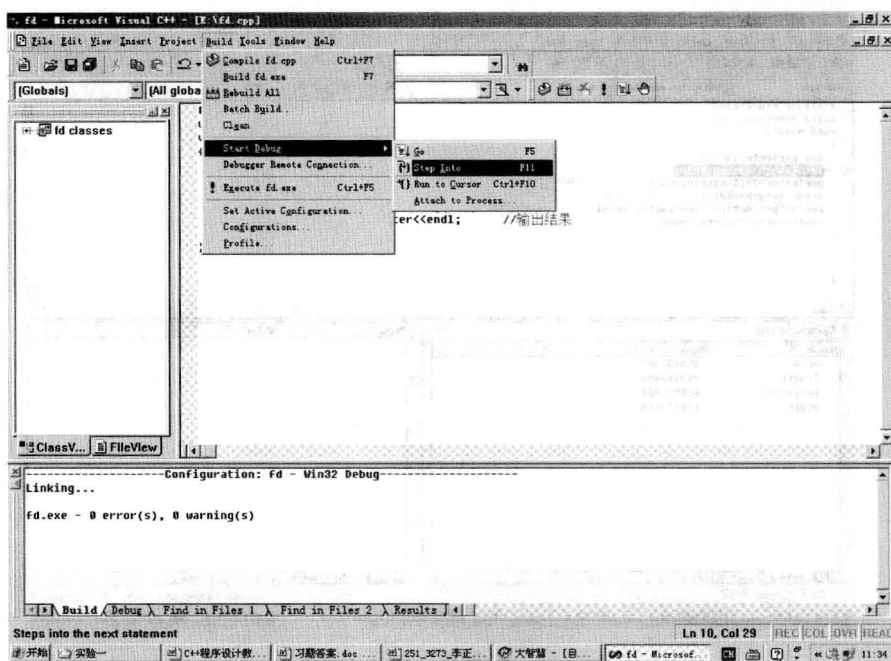


图 1-15 单步执行

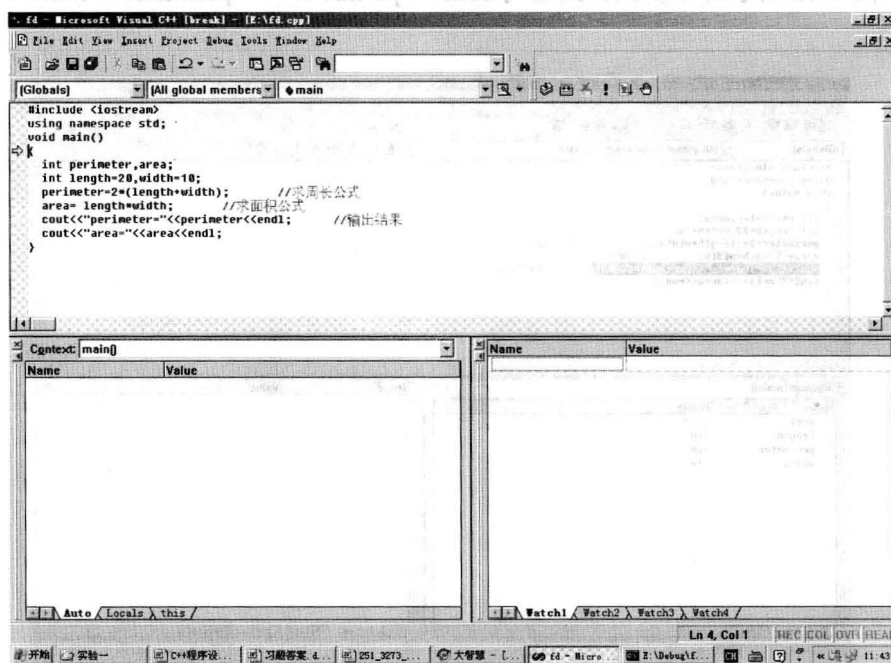


图 1-16 开始单步执行程序

(b) 按 F11 键, 依次执行各条语句, 图 1-17 所示为执行到语句 `int length=20, width=10;` 时各个变量值的赋值情况。