



普通高等教育“十五”国家级规划教材配套辅导书

C

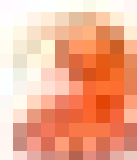
程序设计 实践教程

(第2版)

吴乃陵 李海文



高等教育出版社



清华大学“十二五”普通高等教育本科教材



十十程序设计

实验教程

王 明 著

清华大学出版社

清华大学出版社

普通高等教育“十五”国家级规划教材配套辅导书

C++ 程序设计实践教程(第2版)

吴乃陵 李海文

高等教育出版社

内容提要

本书是与吴乃陵等编著的《C++ 程序设计》(第2版)配套的实践教程。全书分两部分,第一部分是与主教材配套的教学同步实验,共有29个配套实验,以实现精讲多练的教学思想。第二部分是课程设计指导书,讲解MFC编程和面向对象的软件工程相关知识,指导C++的课程设计。

本书可以作为高等学校C++程序设计课程的实验和课程设计指导书,也可供编程爱好者和编程技术人员参考使用。

图书在版编目(CIP)数据

C++ 程序设计实践教程 / 吴乃陵, 李海文. —2 版.

—北京: 高等教育出版社, 2006. 3

ISBN 7-04-018641-1

I. C... II. ①吴... ②李... 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(2006)第 008522 号

策划编辑 何新权 责任编辑 关旭 封面设计 于文燕 责任绘图 朱静
版式设计 张岚 责任校对 王超 责任印制 韩刚

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 高等教育出版社印刷厂

开 本 787×1092 1/16
印 张 18.75
字 数 420 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2003年9月第1版
2006年3月第2版
印 次 2006年3月第1次印刷
定 价 23.60元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 18641-00

前 言

本书是高等教育出版社出版、吴乃陵等编著的《C++ 程序设计》(第2版)教材(书中简称主教材)的配套实践教程。

教育部《关于进一步加强高等学校计算机基础教学的意见》白皮书中指出:应做到主教材和上机实验教材配套,教材内容合理分工。这套教材在合理分工方面走出了新路,面向对象程序设计的教学在主教材中完成,封装了应用程序编程接口的 MFC(标准 Windows)程序设计在实践教程中进行。

本实践教程内容分两部分:

第一部分是与主教材同步的实验指导书。采用 Visual C++ 6.0 集成开发环境下的控制台方式。为了配合精讲多练的教学方法,对实验内容做了大规模的调整与补充,与第2版主教材紧密配合的实验有29个。

第二部分是课程设计指导书。这一部分讲解 MFC 编程和面向对象的软件工程的有关知识,指导 C++ 的课程设计。与第一版相比,删除了 API 编程,第2章改为 MFC 编程基础知识,使学生对 MFC 编程的机理有一个简约而全面的了解,有助于学习 MFC 编程。第3章改动较小,目的是使大学生掌握常用 MFC 类,了解 MFC 程序的体系结构,并能应用 MFC 类进行标准 Windows 程序设计。这两章是课程设计的前导或工具。第4章改动很大,介绍了面向对象软件工程的思想和方法,使用 UML 进行面向对象的系统分析和设计,课程设计采用研究型学习方法,学生在教师指导下分组自选题目,调研并自定内容。这部分内容强调参与开发的全过程,而不是最后的结果。

打“*”号的部分为选读内容。

本书第1、2章由吴乃陵编写,第3、4、5章由李海文编写。全书由吴乃陵统稿。

由于我们水平与能力有限,时间紧迫,尽管是第2版,难免还会有许多错误和不足。欢迎使用和阅读本教程的教师和同学批评指正。编者的电子邮箱是:wunailing@mail.edu.cn。

全部教学资料可从高等教育出版社高等理工教学资源网(<http://www.hep-st.com.cn>)下载,也可以直接向编者索取。

吴乃陵 李海文
2006年1月于东南大学

第一版前言

本实践教材可与吴乃陵等编著的《C++ 程序设计》教材配套使用。我们的思路是:实践教材不仅仅是实验指导书,也是教材的延伸,完成 Windows 下程序设计的初步训练。课堂教学主要完成编程能力的培养中的 C++ 程序设计基础知识和算法(基本编程方法)的学习。本实践教学安排了 20 个与课堂教学同步的上机实验,以巩固所学知识,掌握 C++ 基本编程方法,并指导如何调试程序;还讲解了 Windows 平台下的 MFC 编程方法,同时安排上机实验;最后是课程设计,编制两个小型软件,使大学生初步了解和掌握软件编制的全过程。

Windows 是当前最流行的程序设计工作平台,Microsoft Visual C++ 是最常用的 Windows 平台下的 C++ 程序设计集成环境之一,但 MFC 程序设计的内容并不适合课堂教学,它没有系统的理论知识,所以安排在实践中教学,以自学为主,以实际动手编程为主,更为合适。参照微软的《Mastering MFC Fundamentals》中的要求,我们安排了以下内容:Windows 操作系统介绍,传统的 Windows API 程序设计简介,MFC 程序设计,用户界面资源的应用,对话框的应用,文本和图形输出,文档的保存(序列化),注册表,MDI 应用程序设计以及 5 个同步实验(MFC 4 个)。这只是起步的最低要求。

通常 MFC 教材内容庞杂,学生易迷失在令人眼花缭乱的操作中,而不能了解程序的基本组成。本教材中给出的内容简洁而不失完整性:由应用程序向导建立程序框架,由资源编辑器建立用户界面,由类向导建立成员变量、消息映射和处理函数,加上文档序列化和注册使学生对 MFC 编程的关键步骤一目了然。如要锦上添花,可自学其他 MFC 有关书籍。

作为程序设计课程应该使大学生具有软件工程的概念和软件开发的基础知识。我们在课堂教学部分安排了有关软件工程、软件生命期各阶段的工作任务、软件的质量标准、面向对象的系统分析与设计等内容。最后安排课程设计来实际编写两个小型软件,总共有 10 个管理系统的题目,6 个其他类型,可供选做。

本书第 5 章介绍 Visual C++ 6.0 集成开发环境,供读者在学习过程中查阅。

各部分实验多数安排有范例和编程题,范例供读者学习参考,并要求上机调试;编程题完全由读者独立编制完成,题目较多的可根据情况选做。

为教师提供的资料将在网上公布。

本实践教材与吴乃陵等编著的《C++ 程序设计》教材配套后的全部内容是按本课程的最高要求编写的,但实际上因教学要求不同,后续课程配置不同,可以选用部分内容,甚至只用部分同步实验。

本实践教材的编写大纲由李海文和吴乃陵共同拟定,由李海文编写全书,吴乃陵修改和补充了同步上机实验的内容。

在本书出版之际,感谢东南大学的朱敏教授,她在百忙中审阅了全书,提出了宝贵意见。感谢彭沛教授,课程设计中的管理软件源自他的工作。

因编者水平有限,时间仓促,难免有错误和不足之处,敬请读者指正。

李海文 吴乃陵
2003年5月于东南大学

目 录

第1章 程序设计同步实验指导	1	实验二十五 纯虚函数	90
实验一 Visual C++ 集成开发环境		实验二十六 输入/输出与重载	94
入门	1	实验二十七 文件	95
实验二 简单的 C++ 程序设计	7	实验二十八 异常处理	100
实验三 分支结构程序设计	11	*实验二十九 标准模板库编程	106
实验四 循环结构程序设计	14	*第2章 MFC 编程基础知识	110
实验五 常用算法——枚举法、递推法、 迭代法	17	2.1 传统的 Windows 编程	110
实验六 文本文件的简单应用	19	2.2 MFC 编程	119
实验七 函数的基本概念	21	2.3 Windows 对象句柄	121
实验八 函数的递归算法	23	2.4 MFC 对象与 Windows 对象的 关系	123
实验九 函数的重载和变量的 作用域	25	2.5 MFC 基础类及其层次结构	124
实验十 类与对象的基本概念	27	2.6 MFC 的消息映射	127
实验十一 引用与复制构造函数	32	2.7 MFC 中的文件处理	132
实验十二 运算符重载	34	2.7.1 文档/视图结构	132
实验十三 数组与多维数组	41	2.7.2 存档类序列化	134
实验十四 指针与数组	43	第3章 MFC 程序设计	135
实验十五 模板	46	3.1 MFC 程序设计简介	135
实验十六 排序与查找	50	3.1.1 MFC 定义的宏、全局函数及全局 变量	135
实验十七 模板与类参数	55	3.1.2 MFC 命名规则	136
实验十八 动态内存分配与深复制	57	3.1.3 应用程序框架	136
实验十九 链表及应用	61	3.2 MFC 程序设计	140
实验二十 栈与队列的操作	71	3.2.1 MFC 程序的执行过程	140
*实验二十一 二叉树与二叉排 序树	78	3.2.2 MFC 消息映射	141
实验二十二 继承与派生的基本 概念	81	3.2.3 文档/视图结构及应用实例	144
*实验二十三 多重继承与虚基类	83	3.3 用户界面资源的应用	164
实验二十四 虚函数与多态	88	3.3.1 菜单	164
		3.3.2 工具栏	170
		3.3.3 状态栏	171

3.4 对话框的应用	173	4.2.4 软件开发模型	235
3.4.1 对话框的工作方式、种类和创建方法	173	* 4.3 UML 面向对象系统的模型	236
3.4.2 对话框与程序的连接	177	4.3.1 统一建模语言(UML)简介	236
3.5 基于对话框的应用程序设计		4.3.2 UML 软件开发的常用模型	237
实例	182	* 4.4 UML 面向对象的系统分析和设计	241
3.6 文本和图形输出	186	4.4.1 面向对象的需求分析	241
3.7 文档的保存	192	4.4.2 面向对象的设计	243
3.7.1 文档序列化	192	4.5 软件测试	248
3.7.2 注册表	199	4.5.1 软件测试概念	248
3.8 MDI 应用程序设计	203	4.5.2 软件测试技术	248
实验三十 对话框与控件	212	4.5.3 软件测试策略	252
实验三十一 文档/视图结构及图形与文本输出	214	4.5.4 面向对象的软件测试	254
实验三十二 序列化和文件操作	220	4.6 研究型学习与实施方法	257
实验三十三 多文档与多重视图	226	4.7 课程设计参考选题	259
第4章 课程设计:面向对象系统分析与设计实践	233	第5章 Visual C++ 6.0 集成开发环境介绍	272
4.1 软件的定义	233	5.1 菜单	272
4.2 软件工程	233	5.2 工具栏	285
4.2.1 软件危机和软件工程的提出	233	5.3 项目工作区	286
4.2.2 软件生命期	234	5.4 Visual C++ 向导	287
4.2.3 软件的质量标准	234	5.5 调试工具	288
		参考文献	289

第 1 章 程序设计同步实验指导

实验一 Visual C++ 集成开发环境入门

一、实验目的

1. 初步学会使用 Visual C++ 集成开发环境 (IDE)。① 进入和退出;② 菜单、工具栏的使用;③ 用户窗口区的划分:workspace、source file 和 message。
2. 初步了解调试程序的方法。

二、实验内容

范例:Visual C++ 控制台应用程序设计步骤。

作为学习面向对象的 C++ 的第一步,学习的重点是算法,必须避免用户图形界面的干扰,这时可采用控制台应用程序进行练习。

下面是一个控制台应用程序实例,该程序要求用户从键盘输入 3 个整数,然后按照从小到大的顺序在屏幕上输出。

【步骤一】 进入和退出 Visual C++ 集成开发环境。

启动并进入 Visual C++ 集成开发环境至少有 3 种方法:

(1) 选择“开始”菜单中的“程序”,然后选择 Microsoft Visual Studio 6.0 级联菜单,再选择 Microsoft Visual C++ 6.0 菜单项,如图 1.1 所示。



图 1.1 进入 Visual C++ 集成开发环境

(2) 在桌面上创建 Microsoft Visual C++ 6.0 的快捷方式,直接双击该图标。

(3) 如果已经创建了某个 Visual C++ 工程,双击该工程的 dsw(develop studio workshop)文件图标,也可进入集成开发环境,并打开该工程。

选择 File|Exit 菜单,可退出集成开发环境。

[步骤二] 创建一个控制台应用程序工程。

(1) 进入 Visual C++ 集成开发环境后,选择 File|New 菜单,弹出 New 对话框。单击 Projects 标签,打开其选项卡,在其左边的列表框中选择 Win32 Console Application 工程类型,在 Project name 文本框中输入工程名 Exp1_1,在 Location 文本框中输入工程路径,单击 OK 按钮,如图 1.2 所示。

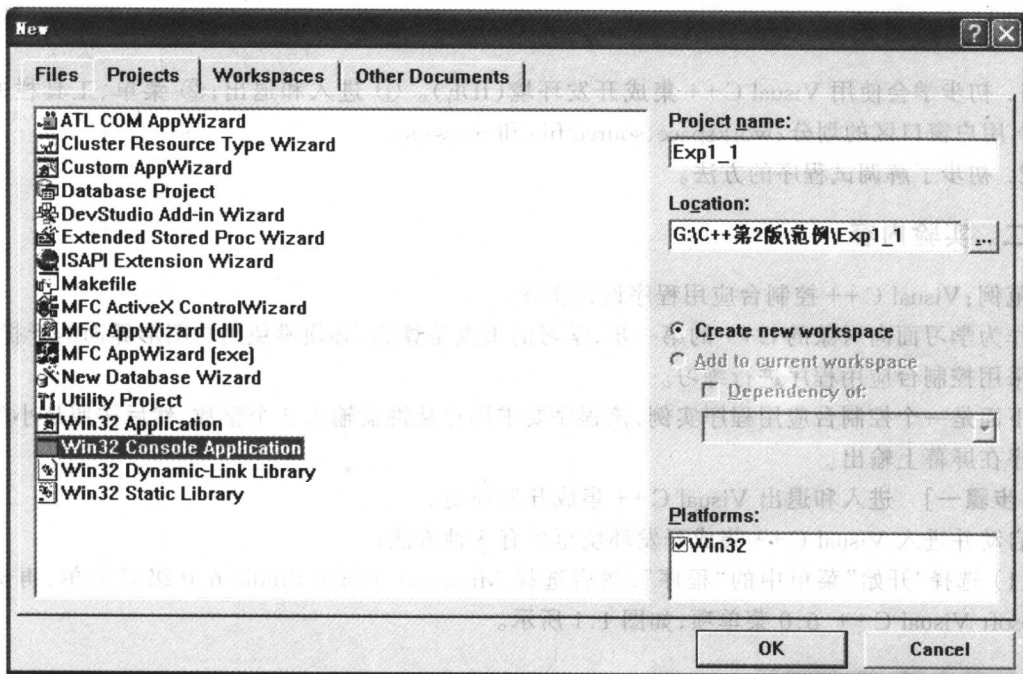


图 1.2 创建新的应用程序界面

(2) 在弹出的对话框(如图 1.3 所示)中,选择 An empty project,单击 Finish 按钮。

(3) 此时出现 New Project Information 对话框,如图 1.4 所示。此对话框中提示用户创建了一个空的控制台应用程序,并且没有任何文件被添加到新工程中,此时,工程创建完成。

[步骤三] 程序的编辑、编译、生成和执行。

(1) 选择 File|New 菜单项,出现 New 对话框,打开 Files 选项卡,在列表框中选择 C++ Source File,在 File 文本框中输入文件名 Exp1_1.cpp,选中 Add to project 复选框,如图 1.5 所示。然后单击 OK 按钮,打开源文件编辑窗口,在其中输入以下源代码:

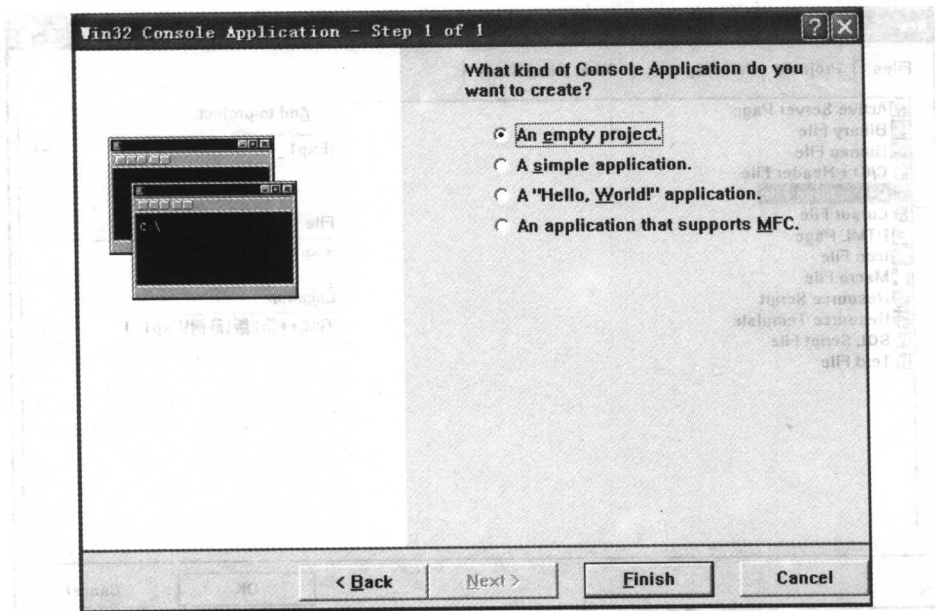


图 1.3 Win32 Console Application Step 1 of 1

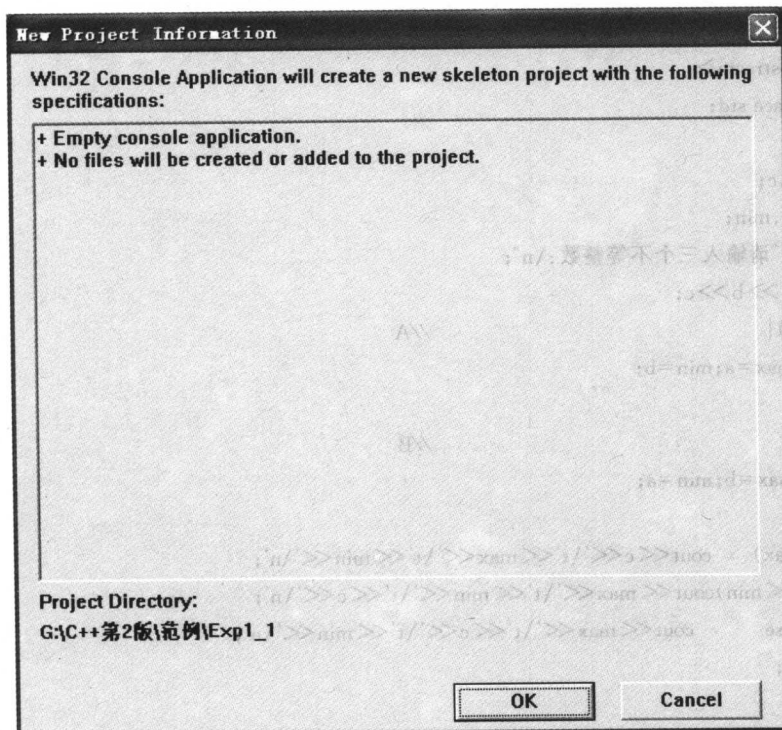


图 1.4 New Project Information 对话框

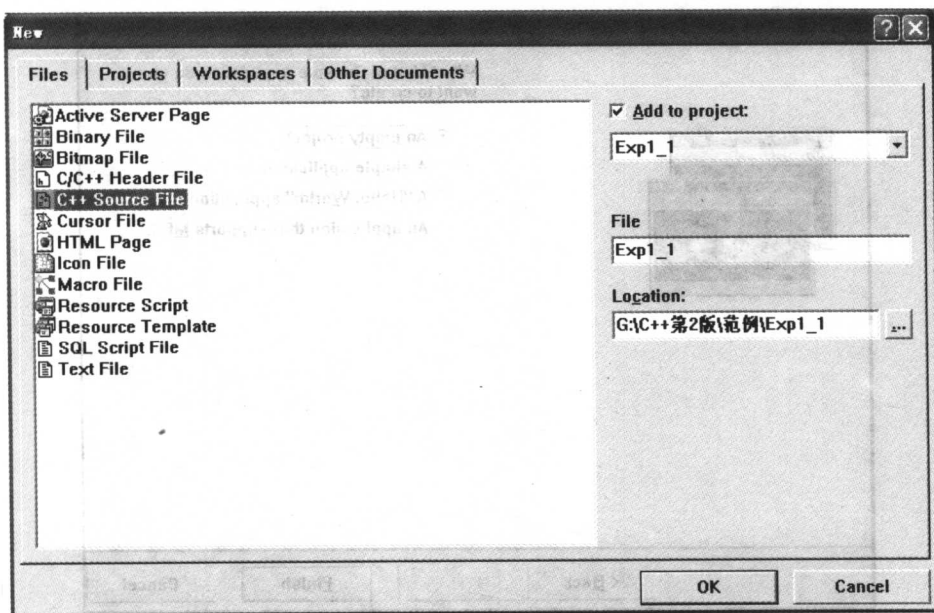


图 1.5 创建新的 C++ 源文件

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    int max,min;
    cout<<"请输入三个不等整数:\n";
    cin>>a>>b>>c;
    if(a>b){                                //A
        max=a;min=b;
    }
    else {                                  //B
        max=b;min=a;
    }
    if(c>max)    cout<<c<<"\t"<<max<<"\t"<<min<<"\n";
    else if(c<min)cout<<max<<"\t"<<min<<"\t"<<c<<"\n";
    else        cout<<max<<"\t"<<c<<"\t"<<min<<"\n";
    return 0;
}
```

//将上述内容保存为 Exp1_1.cpp

对于已经存在的源文件,选择 Project|Add to Project|Files 菜单项,在随后打开的插入文件对话框中选择待添加文件,单击 OK 按钮即可将其添加进工程。

(2) 选择 Build|Compile 菜单项,即可编译源文件 Exp1_1.cpp。系统会在 Output 窗口显示出错(error)信息以及警告(warning)信息。当所有错误都被改正后,可得到目标文件(Exp1_1.obj)。

编译器在 Output 窗口给出语法错误和编译错误信息。

语法错误处理:鼠标双击错误信息可跳转到错误源代码处进行修改,一个语法错误可能引发系统给出很多条 error 信息,因此,发现一个错误并修改后最好重新编译一次,以便提高工作效率。

警告信息处理:一般是触发了 C/C++ 的自动规则,如将一个单精度(浮点)型数据赋给整型变量,需要系统将单精度型数据自动转换为整型,此时小数部分会丢失,因而系统给出警告信息。警告信息不会影响程序执行,本例可以通过强制类型转换去掉警告信息。

(3) 选择 Build|Build 菜单项,连接并建立工程的 EXE 文件,得到可执行文件 Exp1_1.exe。这时编译器可能会给出连接错误(linking error)。产生连接错误的原因可能是缺少所需要的库文件或目标文件,或程序中调用的外部函数没有定义等,只要补充相应文档再重新建立即可。

(4) 选择 Build|Execute 菜单项,执行工程文件,会出现一个类似 DOS 的窗口,按要求输入 3 个不等的整数后按 Enter 键,屏幕上由大到小输出这 3 个整数,如图 1.6 所示。

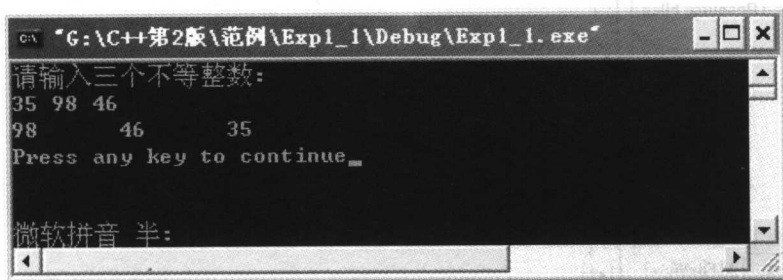


图 1.6 运行结果

[步骤四] 程序的调试。

运行程序时,可能发现程序没有编译错误,而且也能执行,但执行的结果不对,此时,除了仔细分析源程序,还可借助调试工具进行跟踪调试。

例如,若在 Exp1_1.cpp 的 B 行中的 else 后面多加一个分号,即:

```
... else;{max=b;min=a;}
```

使用 88、45、67 这组数据测试此程序,发现输出结果为 67 45 88,显然结果不对。此时需要对程序进行调试,下面介绍调试过程。

首先,在源程序中可能出现错误的行上设置断点,方法是将光标移至该行,然后按 F9 键,或单击工具栏上的手形按钮(再按一次 F9 键或单击手形按钮将取消断点),此时该行左侧出现一

个红色圆点,断点设置成功,如图 1.7 就是在 A 行设置了断点。

选择 Build|Start Debug|Go 菜单命令(也可选择 Build 工具栏上的 Go 图标),程序执行到断点处停止,这时选择 View|Debug Windows 子菜单的 Watch 和 Variables 两个菜单项,打开监视和变量窗口来观察变量值(Watch 和 Variables 窗口的详细介绍见 View 菜单介绍),分析并查找出错原因。

在 Watch 窗口加入 max 和 min 两个变量,进行监视。Watch 窗口的每一行可显示一个变量,其中,左栏显示变量名,双击它可进行编辑;右栏显示变量值。单步执行按 F10(不跟踪进函数)或 F11(跟踪进函数内),尽管 $a > b$, 在执行了 if 后面的 `{max = a; min = b;}` 后,仍然执行了 `{max = b; min = a;}`。当程序执行到箭头所指处时, `max=45`, `min=88`,如图 1.7 所示,与预期结果不相符,说明程序的流程有问题。此时再仔细分析源程序,发现问题在于 else 后多余的分号。

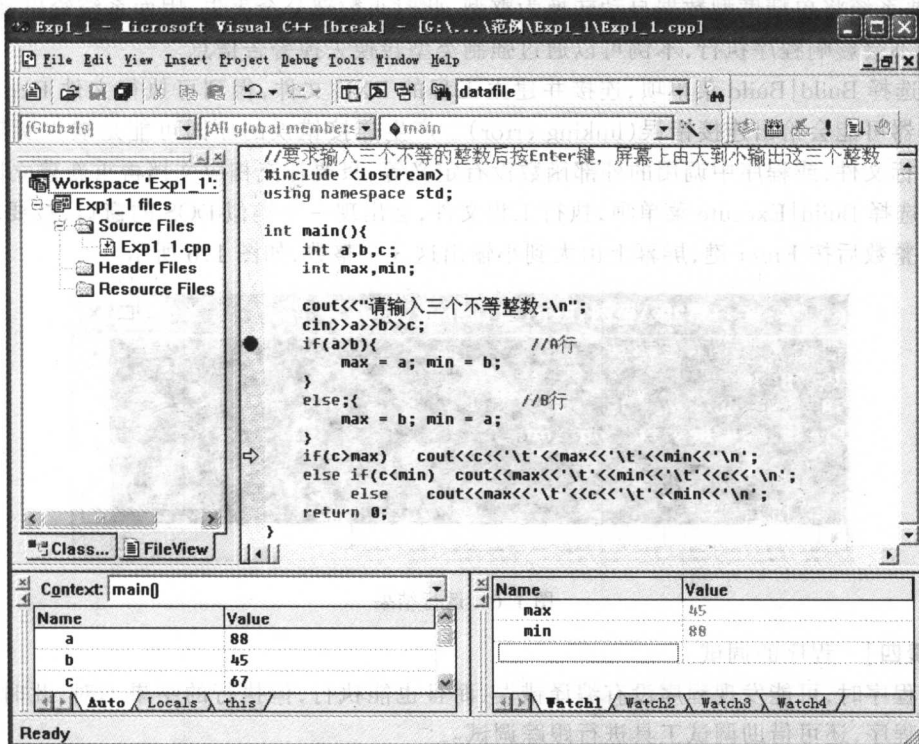


图 1.7 Visual C++ 集成开发环境及程序的调试

调试过程中,Variables 窗口动态显示各变量值随程序执行而变化的结果。在面向对象程序设计中,若程序中有类的对象,Variables 窗口的 this 页面可显示当前 this 指针所指向对象的各个值。

经过反复的修改和调试,使程序中所有问题都得到改正后,可得到正确的执行结果。

实验二 简单的 C++ 程序设计

一、实验目的

1. 掌握 C++ 程序的基本结构。
2. 掌握 C++ 基本数据类型与运算符。
3. 熟悉输入/输出方法。

二、实验内容

1. 范例:输入两个整数,将它们交换后输出。

[分析一] 要交换两个整数,一般是使用一个中间变量经过几次赋值实现。

(1) 进入 Visual C++ 开发环境,选择 File|New 菜单,弹出 New 对话框,打开 Projects 选项卡,在列表框中选择 Win32 Console Application 工程类型,在 Project name 文本框中输入工程名 Exp2_1,选择合适的路径,单击 OK 按钮。

(2) 在随后的对话框中选择 An empty project,单击 Finish 按钮。

(3) 选择 File|New 菜单项,在 New 对话框的 Files 选项卡中选择 C++ Source File,输入文件名 Exp2_1.cpp,选中 Add to project 复选框,单击 OK 按钮,打开源文件编辑窗口,输入以下源代码:

```
#include<iostream>
using namespace std;
int main(){
    int a=-3,b=7,temp;
    cout<<"a="<<a<<"\t"<<"b="<<b<<endl;
    temp=a;
    a=b;
    b=temp;
    cout<<"a="<<a<<"\t"<<"b="<<b<<endl;
    return 0;
}
```

*[分析二] 采用按位异或方法,因为任一位数与 0 异或保持不变,而与 1 异或后翻转(0 变为 1,1 变为 0),可编写程序如下:

```
v1=a;v2=b;           //原 v1 为 a,v2 为 b
v1=v1^v2;             //即 v1 为 a^b,v2 保持不变
v2=v1^v2;             //即 v2=a^b^b=a
v1=v1^v2;             //即 v1=a^b^a=b,交换成功
```

```
#include<iostream>
using namespace std;
int main(){
    int a=-3,b=7,temp;
    cout<<"a="<<a<<"\t"<<"b="<<b<<endl;
    a=a^b;
    b=a^b;
    a=a^b;
    cout<<"a="<<a<<"\t"<<"b="<<b<<endl;
    return 0;
}
```

[要求]

(1) 仔细阅读程序,掌握程序结构,找到程序的数据定义部分和操作部分,辨识各部分功能,注意输入/输出语句的使用。

(2) 按照正确的步骤进入 Visual C++ 集成开发环境,在自己创建的工程中录入上述源程序。注意书写格式,养成良好的编程习惯;消除语法错误,编译、连接程序。

(3) 运行程序,输入数据观察结果,并进行测试。

* (4) 用位运算符重新运行程序并对比运行结果。

2. 编写满足下列要求的程序:根据输入的球的半径,分别计算球的表面积、体积和质量(假设球的密度为 7.8 kg/dm^3),并输出计算结果。

[提示] 球表面积计算公式为 $s = 4\pi r^2$,球体积计算公式为 $v = \frac{4}{3}\pi r^3$ 。

注意输入/输出形式,要求输入前应有提示性输出,如 "Please Input the Radius of the Ball", 注意变量的命名,最好能见文知义,如 Radius、Volume、Weight 等。另外,由于整数相除取整,注意在计算公式中使用正确的变量数据类型。

3. 范例:用 sizeof 运算符计算 C++ 中 char、short、int、long、float、double 等基本数据类型所占的字节数,并按以下格式输出:

```
sizeof(char)=1 byte
sizeof(short)=2 bytes
sizeof(int)=4 bytes
...
```

再将计算结果以表格形式输出:

char	1
short	2
int	4
...	