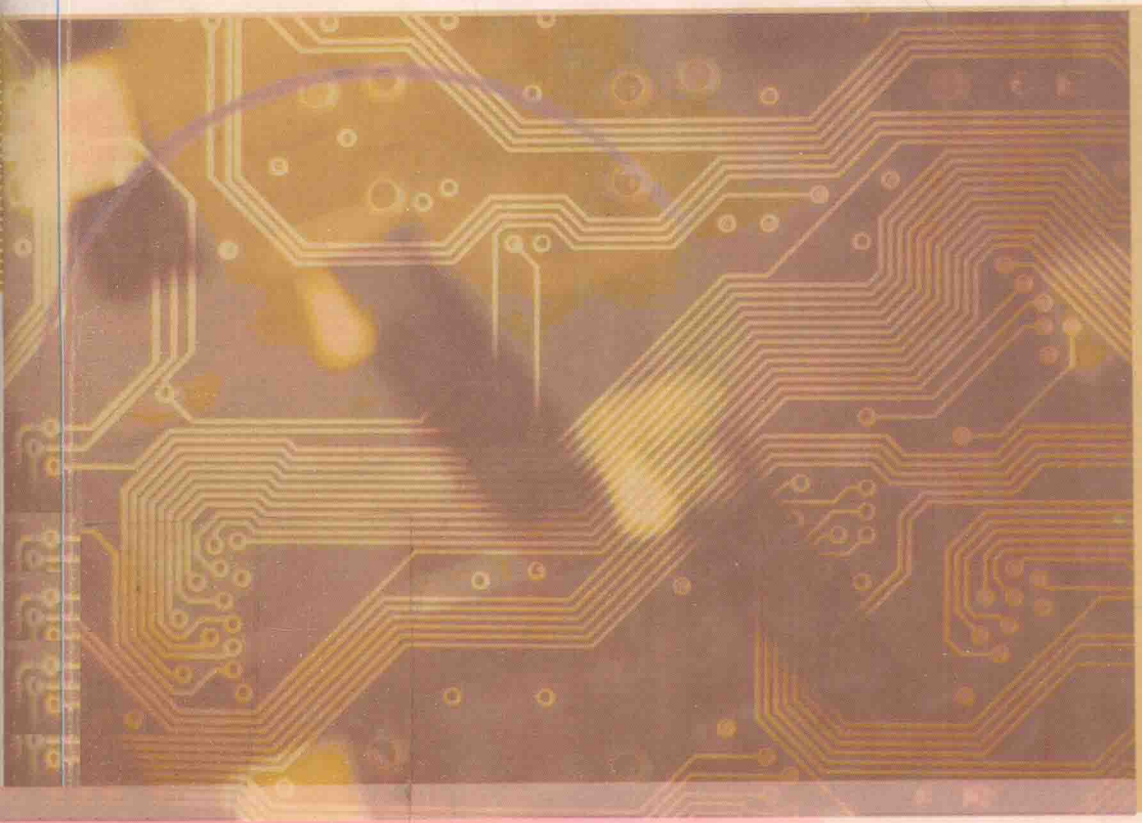
教育部人才培养模式改革和开放教育试点教材

面向对象程序设计实验

徐孝凯　等编



中央广播电视大学出版社

教育部人才培养模式改革和开放教育试点教材

面向对象程序设计实验

徐孝凯　等编

中央广播电视大学出版社

图书在版编目（CIP）数据

面向对象程序设计试验/徐孝凯等编 .—北京：中央广播电视大学出版社，2003.7

教育部人才培养模式改革和开放教育试点教材

ISBN 7-304-02404-6

Ⅰ.面… Ⅱ.徐… Ⅲ.面向对象语言—程序设计—电视大学—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2003）第 065839 号

教育部人才培养模式改革和开放教育试点教材

面向对象程序设计实验

徐孝凯 等编

出版·发行/中央广播电视大学出版社

经销/新华书店北京发行所

印刷/北京市银祥福利印刷厂

开本/787×1092 1/16 印张/19 字数/471 千字

版本/2003 年 6 月第 1 版 2003 年 7 月第 1 次印刷

印数/0001—11000

社址/北京市复兴门内大街 160 号 邮编/100031

电话/66419791 68519502 （本书如有缺页或倒装，本社负责退换）

网址/http://www.crtvup.com.cn

书号：ISBN 7-304-02404-6/TP·182

定价：25.00 元

前　言

面向对象程序设计课程是全国电大系统计算机科学与技术专业一门专业基础课，该课程主教材为《C++面向对象程序设计》，由王萍主编，清华大学出版社出版。本书是与之配套的实验教材。

面向对象程序设计课程以 Visual C++6.0 语言环境为依托，介绍 C++语言的基本语法知识和面向过程与面向对象的程序设计方法，为后续开设的数据结构课程奠定基础。在该课程的教学过程中，上机实验教学是一个非常重要的、必不可少的环节，它同课堂讲授相辅相成，共同完成教学任务，提高教学质量。

由于该课程没有其他相配套的学习辅导教材，所以在实验内容的安排上，有的地方对主教材中缺少的内容(如模块、文件、多态性等)进行了补充，有的地方对主教材中已有的内容进行了辅导和深化。因此，既要把本书作为实验教材来使用，又要把它作为学习辅导书来使用。

本书共包括 10 个实验，每个实验都包括实验目的、实验预备知识、实验内容和实验要求四个部分。实验目的给出通过该实验所能达到的预期效果，实验预备知识给出完成该实验所需要涉及和掌握的主要知识，它是对主教材有关内容的辅导，实验内容给出该实验的一些上机调试程序和编程题目，实验要求给出做好本次实验的具体任务和明确要求。本书附录给出该课程综合练习题及参考解答。

要求每个学生至少完成 8 个实验，当然都完成更好。

为了学好面向对象程序设计这门课程，加强上机实验非常重要，这仅仅依靠本实验教材中所给的实验是远远不够的，有条件和有兴趣的学生应该针对学习和实践中出现或思考的问题，主动地编写程序并上机调试和运行，主动地尝试和使用 C++操作界面所提供的各种功能，从而提高自己分析问题和解决问题的能力。

本教材中的所有程序都具有良好的结构化和可读性，都在 Visual C++6.0 环境下调试运行通过，确保它们是正确无误的。

本书由清华大学计算机科学与技术系教授唐龙审定，唐教授在十分繁忙中本着对电大教学高度负责的精神，认真仔细地审阅了全部书稿，提出了宝贵的意见，提高了本书的质量，在此谨向唐教授深表敬意和感谢。

本书除了在电大系统使用外，还可以作为普通高校开设 C++语言程序设计课程的实验教材或学习参考书。

由于本人水平有限，书中难免出现一些不足或错误，敬请广大师生批评指正，本人不胜感谢。

徐孝凯

2003 年春

目　　录

实验一　了解 C++ 运行环境

一、实 验 目 的

1. 了解 VC++6.0 开发环境。

2. 会使用 VC++6.0 开发环境输入、编辑、编译、连接和运行一个 C++ 程序。

3. 能够进一步掌握 C++ 程序的结构。

4. 体会标准输入流 cin 和标准输出流 cout，以及相应的提取操作符 >> 和插入操作符 << 的作用。

5. 初步了解变量定义语句、变量赋值语句、条件语句、返回语句等的格式和含义。

6. 初步了解函数定义格式，以及函数调用的格式与作用。

二、实验预备知识

本课程采用 Microsoft Visual C++6.0 为蓝本介绍 C++ 语言的知识，因此实验教材也将使用这个 C++ 运行环境。

Microsoft Visual C++ 6.0 简称 VC++ 6.0，它是美国微软（MicroSoft）公司研制开发的 C++ 语言版本，是一个集 C++ 程序编辑、编译、调试、运行和在线帮助等功能及可视化软件开发功能为一体的软件开发工具，或称开发环境、开发系统等。由于此开发环境功能强大，内涵丰富，需要一个循序渐进的过程逐步熟悉和掌握，开始只要求同学们了解和掌握如何编辑、编译和运行一个 C++ 控制台应用程序（console application program）即可。其他方面的使用需要同学们借助“在线帮助系统”和查阅该开发环境的有关使用说明及参考资料才能实现。

在VC++6.0集成开发环境（界面）中，要建立一个 C++ 控制台应用程序（以后简称 C++ 程序），首先要建立一个项目（project），就是在计算机的外存磁盘上建立一个表示该项目的一个专用目录，然后把一个 C++ 完整程序作为一个或若干个文件保存到这个目录中，再通过编译、连接、运行等步骤实现程序所具有的功能。

用 C++ 语言编写出一个完整的程序后，第一步需要上机建立相应项目并输入和编辑该程序所含的一个或多个程序文件以及一些头文件，其中有一个程序文件必含有主函数，被称之为程序主文件，简称主文件，它通常被首先输入和编辑；第二步对每个程序文件进行编译生成各自的目标代码（即二进制代码）文件，通常主文件被首先编译并生成主目标文件，注

意用户建立的头文件不需要编译；第三步使主目标文件与同一程序中的其他目标代码文件以及有关 C++ 库函数文件相连接，生成一个可执行（即运行）文件；第四步运行最后生成的可执行文件，实现用户编程所需要的对数据的计算或处理功能。

当由一个程序文件编译生成一个目标文件时，目标文件的主名与程序文件的主名相同，而扩展名改为 .obj，当把程序中的所有目标文件连接生成一个可执行文件时，该文件的主名采用项目名，扩展名为 .exe。

上机输入和运行程序的操作过程如下：

1. 输入和编译程序

用户在第一次上机使用VC++6.0集成开发环境时，若在机器上还没有安装VC++6.0语言版本的软件，则应设法找到该软件光盘并通过光盘驱动器按照系统给出的安装信息和步骤安装到机器上，然后才能使用它。

若在 Windows 操作系统桌面上含有以 Microsoft Visual C++ 6.0 或 Msdev 为名字的图标  ，则双击后就启动该软件，在屏幕上打开VC++6.0集成开发环境操作界面窗口，否则，应单击屏幕左下角的开始按钮打开开始菜单，接着从中单击程序菜单项打开程序菜单，从中单击 Microsoft Visual Studio 6.0 菜单项打开该菜单，再接着单击或双击 Microsoft Visual C++6.0 菜单项运行该程序，则就打开了相应的操作界面窗口，进入了VC++6.0集成开发环境。

打开的VC++6.0操作界面如图 1-1 所示，其中最顶行为窗口标题行，将显示出当前编辑的程序文件的文件名，第二行为菜单行，其中每一个菜单项都对应一个下拉式菜单，菜单中的每一个菜单项都是一条操作命令，都具有一定的操作功能，第三行为按钮工具行，向下左半部为工作区窗口，右半部为程序编辑窗口，整个操作界面的最下部为状态输出窗口。

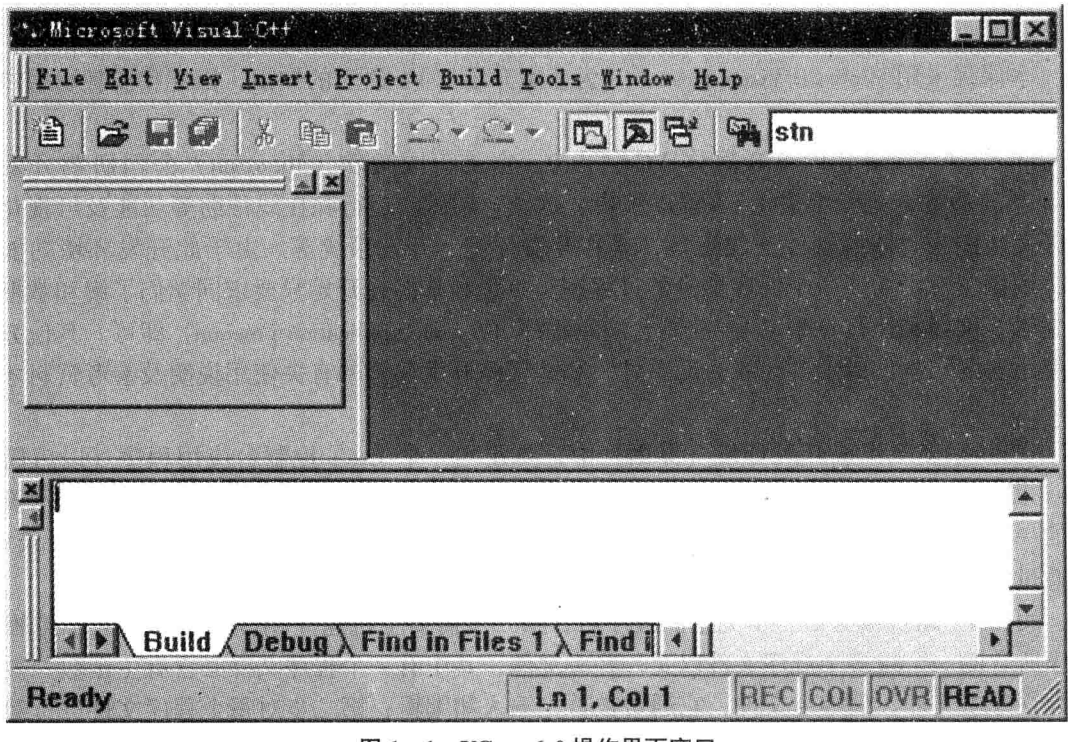


图 1-1　VC++6.0 操作界面窗口

为了首先建立一个程序项目，然后再输入程序，请单击菜单行中的 File 菜单项打开此下拉菜单，从中选择 New 菜单项得到如图 1－2 所示的 New 对话框。

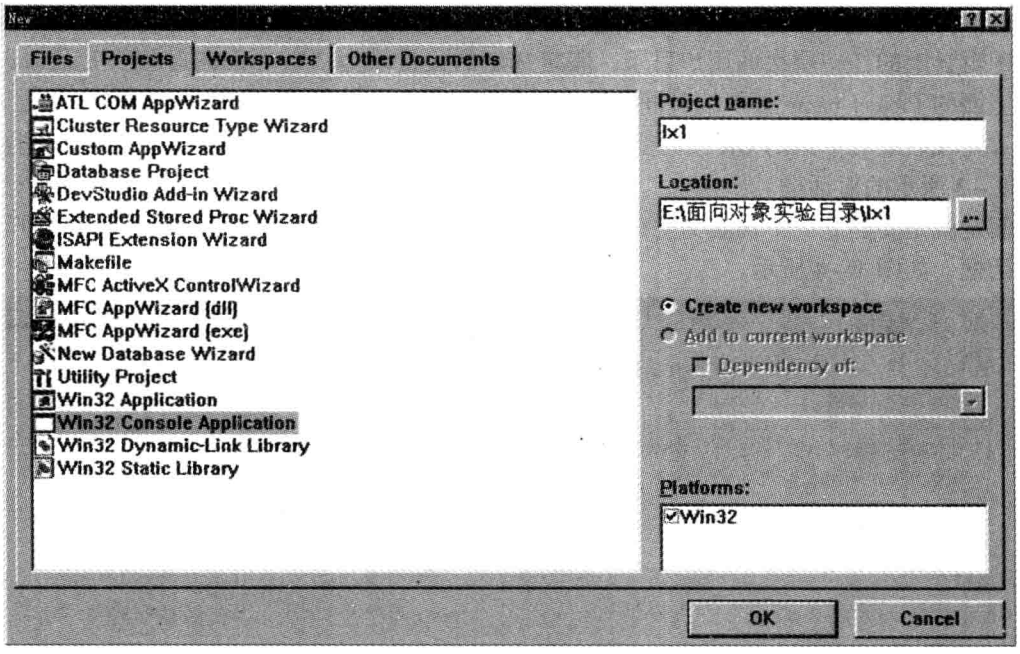


图 1－2　建立项目的 New 对话框

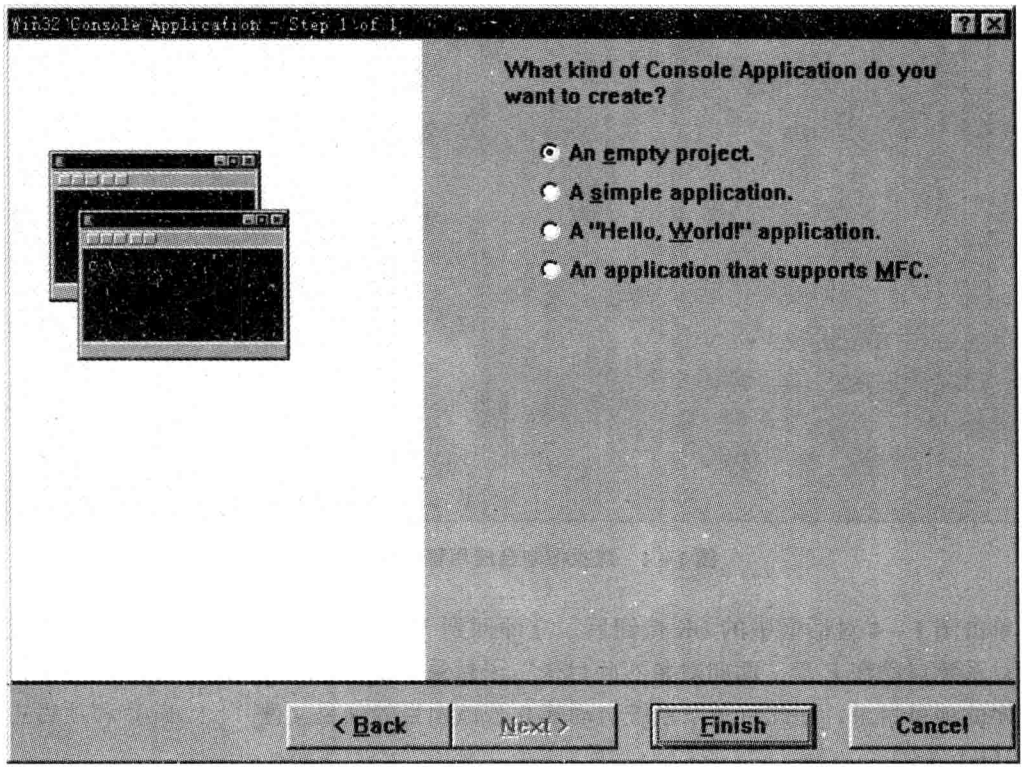


图 1－3　建立控制台应用程序的第一步

从打开的 New 对话框中选择 Projects 标签，再从该标签中选择 Win32 Console Application 菜单项，准备建立一个控制台应用程序的项目目录，再对右边的 Location 文本编辑框中输入或通过该框右边按钮选择一个待建立的项目目录的父目录，它可以是逻辑盘的根目录，也可以是逻辑盘上的任一层次的用户目录，假定选定的父目录为“E:\面向对象实验目录”，再对其上面的 Project name 文本编辑框中输入一个标识符作为项目目录名，例如输入 lx1，项目名输入完成后最右下角的 OK 按钮变为有效，单击该按钮就关闭了该对话框，在屏幕上显示如图 1-3 所示的对话框。

假定就是要建立一个初始为空的项目，用于保存用户程序则单击 Finish 按钮，打开下一个对话框，如图 1-4 所示。

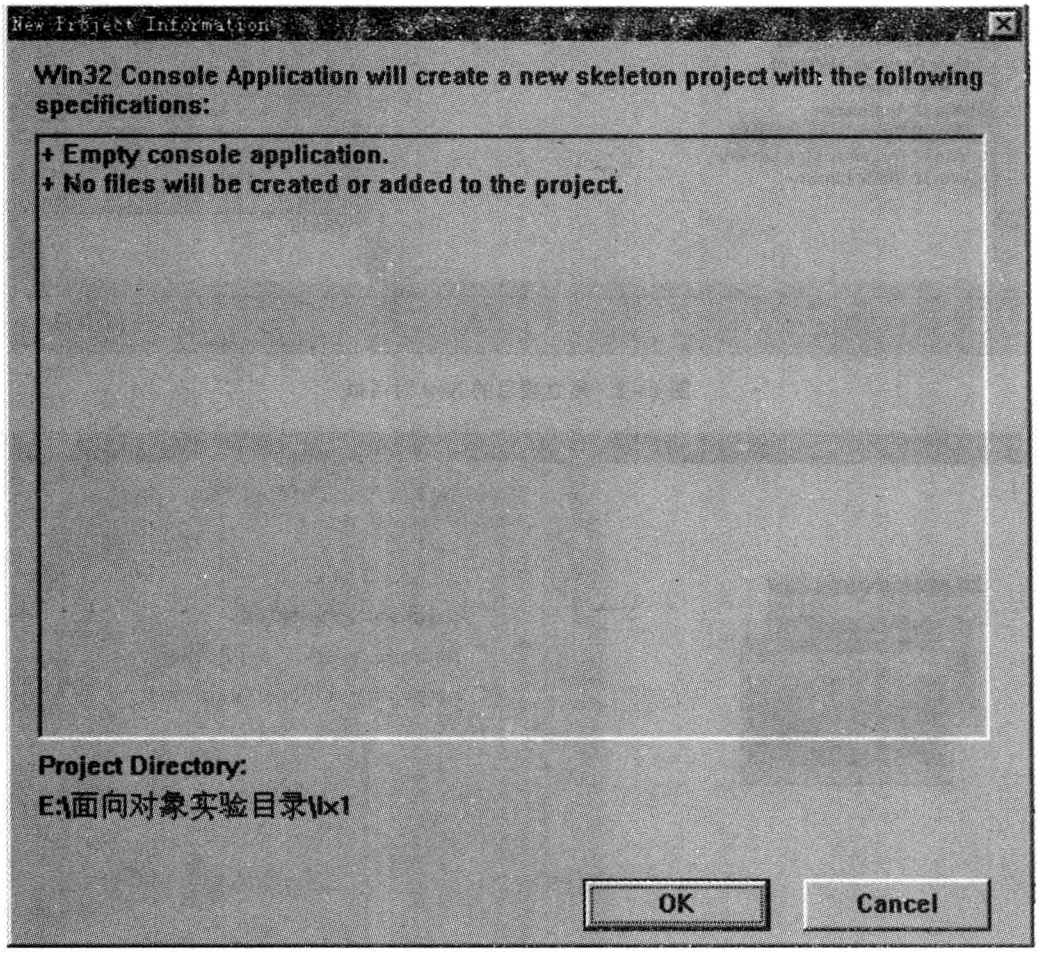


图 1-4　建立控制台应用程序的第二步

单击图 1-4 对话框中的 OK 按钮后，立即回到 VC++ 集成开发环境界面中。通过以上步骤，系统自动在 E 盘“面向对象实验目录”子目录下建立了一个 lx1 子目录，用它作为待建程序的项目目录，在此目录下系统自动建立了该项目的项目文件（lx1.dsp）和项目工作区文件（lx1.dsw）。

回到 VC++ 集成开发操作界面后，为了建立项目（即整个程序）中的每个程序文件和

头文件，需要单击菜单行中的 File 菜单项打开此下拉菜单，从中选择 New 菜单项则打开 New 对话框，再从该对话框中选择 Files 标签，如图 1－5 所示。

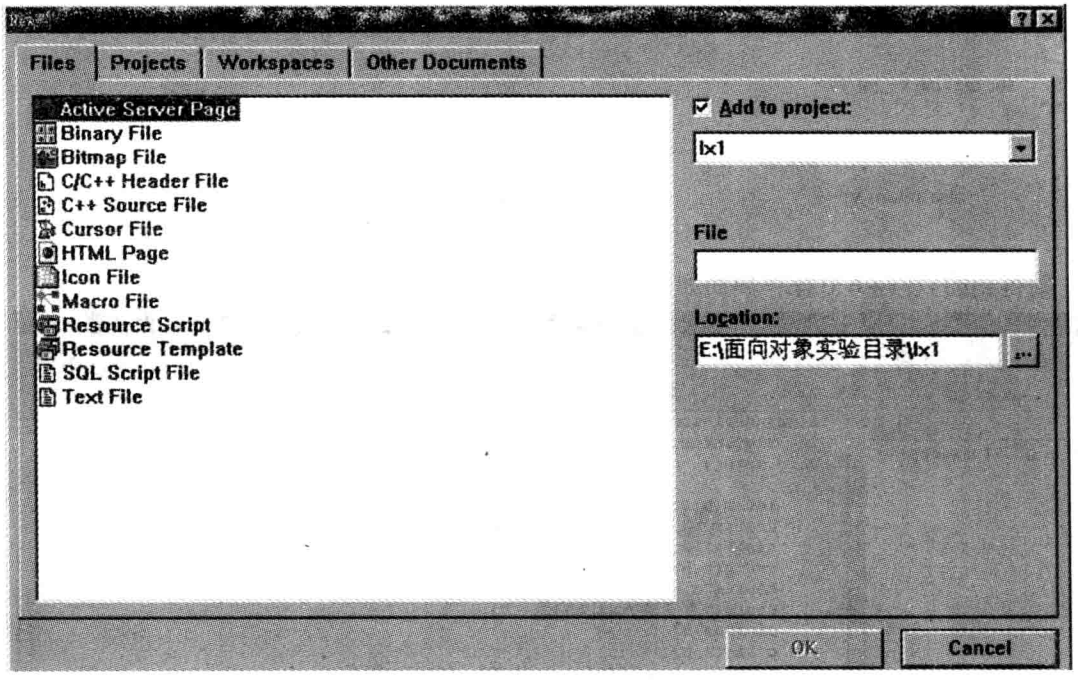


图 1－5　建立文件的 New 对话框

从 File 标签的下拉菜单中若选择 C/C++ Header File 菜单项则可新建一个 C++ 头文件，若选择 C++ Source File 菜单项则可新建一个 C++ 源程序文件。假定选择 C++ Source File 菜单项建立一滚 C++ 程序文件，接着在 New 对话框右边的 File 文本编辑框中输入一个新建文浸的文件名，缺省扩展名为 .cpp，假定输入的程序文件名为 samp1，最后单击右下角的 OK 按钮则就关闭了该对话框，回到 VC++ 集成开发环境界面。此时系统就在 E:\面向对象实验目录\lx1 目录下建立了一个空的 samp1.cpp 文件，待用户通过 VC++ 集成开发环中的程序编辑窗口输入和编辑该程序文件的内容。假定把下面的程序作为 samp1.cpp 文件进行输入和编辑。

```
#include<iostream.h>
int big(int x, int y);
void main()
{
    int a, b, c;
    cout<<"输入 a 和 b 的值:";
    cin>>a>>b;
    c=big(a, b);
    cout<<"a, b, c="<<a<<','<<b<<','<<c<<endl;
    cout<<"重新输入 a 和 b 的值:";
    cin>>a>>b;
```

```
    c = big(a, b);
    cout << "a, b, c = " << a << ',' << b << ',' << c << endl;
}

int big(int x, int y)
{
    if (x >= y) return x;
    else return y;
}
```

则得到的VC++6.0操作界面如图1-6所示。

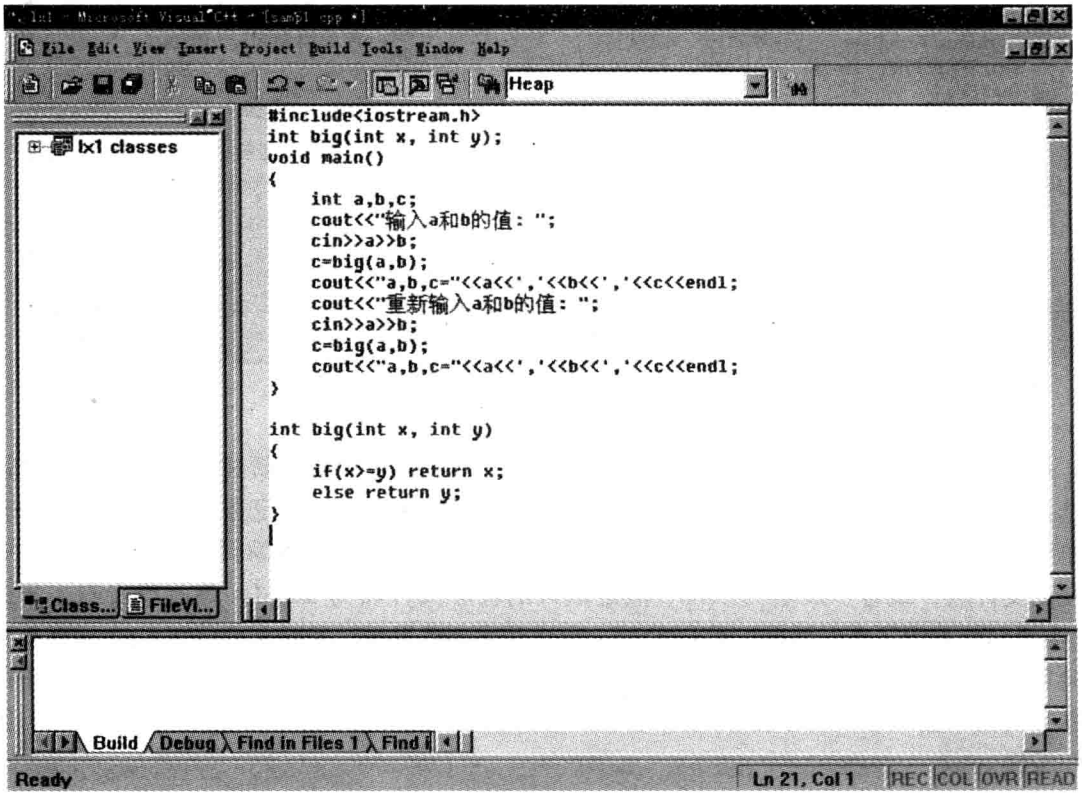


图1-6　工作区窗口和编辑窗口非空的C++操作界面

2. 编译程序文件

当输入和编辑好一个程序文件后，接着编译它。此时从菜单行中选择Build菜单项下拉出该菜单，从中单击第一个菜单项Compile，即可编译在编辑窗口中打开的程序文件，生成一个扩展名为.obj的目标文件。程序主文件通常被首先编译，然后再编译其他程序文件。

一个程序文件的编译过程是：首先处理所有预处理命令，如对于预处理包含命令，将把它替换为所包含的头文件或程序文件的内容（使用#include命令也可以包含扩展名为.cpp的程序文件），接着删除掉所有注释内容（实际上是用一个空格取代），然后按行从上向下进行语法分析，最后生成相应的目标文件。

若在编译过程中检查出语法错误，则将在状态输出窗口显示出产生错误的程序行行号和

错误原因，以便用户重新回到编辑窗口修改错误。

编译时检查出的错误包含两类：一类为严重错误（error），又称为致命错误，用户必须修改它，否则不能进一步向下处理；另一类为警告错误（warning），它不影响进入下一步处理过程，但最好把它修改掉，使得程序在编译后不产生任何错误。

当编译完一个程序文件后，将在状态输出窗口显示出各类错误的个数，若不出现任何错误则显示出“0 error(s)，0 warning(s)”信息，表示没有错误。

若你输入和编辑有多个程序文件和头文件，则它们的文件名将出现在菜单行上的Window菜单项的下拉式菜单的底部，通过选择任一个文件名将使它成为当前编辑文件，接着可对它进行编辑和编译。

注意：带扩展名为 .h 的头文件不能被编译。

3．连接程序文件

连接程序文件就是将一个程序中的主目标文件与其他目标文件和相关的库函数目标文件连接起来形成一个可执行的文件。具体连接操作是：从菜单行上单击 Build 菜单项下拉出相应菜单，接着从中单击第二个菜单项 Build 即可。若连接过程没有发现任何错误，则表示连接成功，此时在状态输出窗口显示出“0 error(s)，0 warning(s)”信息，若连接过程发现有错误，则将在状态输出窗口显示出发生错误的文件、所在的行号和出错原因，用户应根据这些信息修改有关程序文件中的错误，然后再重新进行编译和连接。

4．运行程序

运行程序就是运行对该程序编译和连接而生成的可执行文件。具体操作是：从菜单行上单击 Build 菜单项下拉出相应菜单，从中单击 Execute 菜单项即可。程序运行时将自动打开一个输出显示窗口，并且使显示光标处于该窗口左上角位置，每次执行程序中 cout 语句输出的内容和执行 cin 语句从键盘输入的内容都将从当前光标位置起显示出来，然后显示光标自动后移，即移到所显示内容的后面，再向显示器输出的内容将接着向后显示出来。程序在运行结束前，将在该窗口自动显示出“Press any key to continue”提示信息，用户按下任一键后将关闭该窗口，重新回到 VC++6.0 操作界面窗口。

当对一个程序调试结束后，打开 C++ 主窗口（即 C++ 集成操作界面）上的 File 菜单，从中选择 Close Workspace 菜单项，就关闭了该项目的工作区。接着你可以建立新的程序项目，处理另一个程序，或者从 File 菜单中选择 Open Workspace 菜单项，从打开的该对话框中查找出一个已有项目目录，从该目录中选择带扩展名为 .dsw 的工作区文件，以便重新修改和运行该项目中的程序。

三、实 验 内 容

1．
```
# include < iostream.h >
void main( )
{
    int x, y;
    x = 5; y = 6;
```

```
    cout << "x + y = " << x + y << ',';
    cout << "x * y = " << x * y << endl;
}
```

2.
```
#include < iostream.h >
int cube(int);
void main(void)
{
    cout << "cube(3) = " << cube(3) << endl;
    cout << "cube(5) = " << cube(5) << endl;
    cout << "cube(8) = " << cube(8) << endl;
}
int cube(int x)
{
    return x * x * x;
}
```

3.
```
#include < iomanip.h >
#include"abc.cpp"
void main()
{
    double a, b, c;
    double averageValue;
    a = 2; b = 3; c = 4;
    averageValue = AVE(a, b, c);
    cout << "averageValue:" << averageValue << endl;
    averageValue = AVE(a, b + 1, c + 2);
    cout << "averageValue:" << averageValue << endl;
}
```

其中 abc.cpp 文件的内容如下：

```
double AVE(double x, double y, double z)
{
    return (x + y + z) /3;
}
```

4.
```
#include < iostream.h >
#include "example.h"
void main()
{
    int a, b, c;
    cout << "请输入三个整数:";
    cin >> a >> b >> c; //任意给出三个整数即可
    cout << "最大值:" << max _ value(a, b, c) << endl;
```

```
    cout << "最小值:" << min_value(a, b, c) << endl;
}
```

其中 example.h 文件的内容如下：

```
int max_value(int a, int b, int c);
int min_value(int a, int b, int c);
```

这两个函数的定义（又称为函数的实现或具体实现）被保存在另一个程序文件中，它将被编译后连接到主文件中产生出可执行文件。该程序文件的内容如下：

```
int max_value(int a, int b, int c)
{
    if (a < b) a = b; //若 a 小于 b 则将 b 的值赋给 a
    if (a < c) a = c; //若 a 小于 c 则将 c 的值赋给 a
    return a;
}
int min_value(int a, int b, int c)
{
    if (a > b) a = b; //若 a 大于 b 则将 b 的值赋给 a
    if (a > c) a = c; //若 a 大于 c 则将 c 的值赋给 a
    return a;
}
```

四、实 验 要 求

1. 为上述实验内容中的每个程序分别建立相应的程序项目并调试运行，记录每个程序的运行结果并分析其正确性。

2. 自己编写一个多文件结构的简单程序，并上机调试和运行，以及分析运行结果。

实验二　数据类型和表达式

一、实 验 目 的

1. 熟悉 C++ 语言中预定义的数据类型的表示和含义。
2. 熟悉 C++ 语言中各种常量的表示以及枚举类型和枚举常量的定义。
3. 熟悉 C++ 语言中变量的定义和表示以及符号常量的定义和表示。
4. 熟悉 C++ 语言中各种运算符的表示和含义，各种表达式的构成和运算。
5. 熟悉 C++ 语言中常用数学函数的表示和含义。
6. 进一步熟悉 VC++6.0 操作环境。

二、实验预备知识

2.1　数据类型

数据是人们记录事件和事物的符号表示。如记录人的姓名用汉字表示，记录人的年龄用十进制数字表示，记录人的体重用十进制数字和小数点表示等，由此得到的姓名、年龄和体重都叫数据。根据数据的性质不同，可以把数据分为不同的类型。在日常使用中，数据主要被分为数值和文字（即非数值）两大类，数值又细分为整数和小数两类。

在 C++ 语言中，数据分类如图 2-1 所示。

图 2-1 中每一种无法再分解的数据类型为 C++ 中的一种具体类型。每一种具体类型都对应着惟一的类型关键字、类型长度和值域范围，见表 2-1 所示。但对于结构性的数据类型，其类型长度和值域范围视具体情况而定，无法笼统给出。

下面对表 2-1 作几点说明：

（1）在每一种类型的关键字一栏中，用逗号分开的各组关键字是等价的，都是表示该类型的关键字。如 int 和 signed int 都表示有符号整数类型。

（2）整数类型简称整型。大的整数类型包括小的整数类型、字符类型、逻辑类型和枚举类型，而小的整数类型又包括短整型（short int）、整型（int）和长整型（long int）三种具体类型。读者应根据上下文联系来理解以后叙述中所用“整型”的含义。

（3）对于每一种整数类型和字符类型，又可分为有符号和无符号两种类型。通常使用较多的是有符号类型，所以时常也把有符号类型简称为所属类型。如把有符号整数类型简称为

整型或 int 型，把有符号字符类型简称为字符型或 char 型。

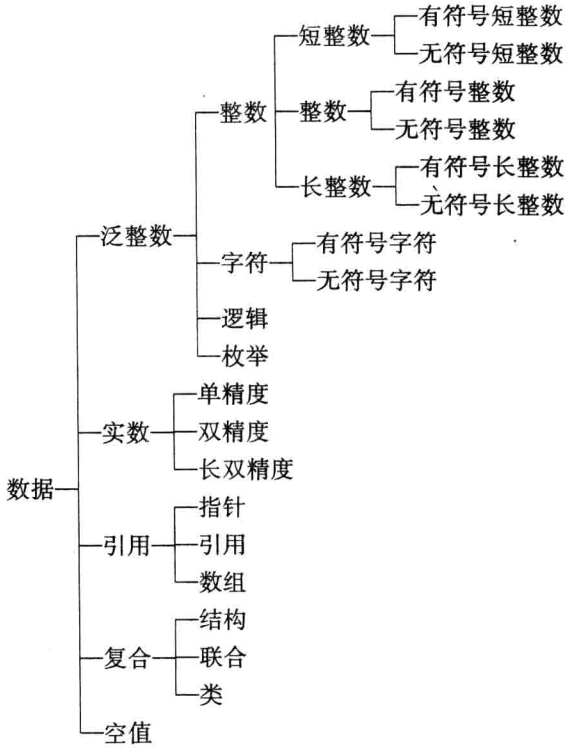


图 2-1　C++ 数据分类

表 2-1　C++ 数据类型

类型	关键字	长度	值域范围
有符号短整数	short，short int，signed short int	2	$-2^{15} \sim 2^{15}-1$ 内的整数
无符号短整数	unsigned short，unsigned short int	2	$0 \sim 2^{16}-1$ 内的整数
有符号整数	int，signed int	4	$-2^{31} \sim 2^{31}-1$ 内的整数
无符号整数	unsigned，unsigned int	4	$0 \sim 2^{32}-1$ 内的整数
有符号长整数	long，long int，signed long int	4	$-2^{31} \sim 2^{31}-1$ 内的整数
无符号长整数	unsigned long，unsigned long int	4	$0 \sim 2^{32}-1$ 内的整数
有符号字符	char，signed char	1	-128 ~ +127 内的整数
无符号字符	unsigned char	1	0 ~ 255 内的整数
逻辑	bool	1	0 和 1
枚举	enum < 枚举类型名 >	4	为 int 值域内的一个子集
单精度数	float	4	$-3.402823 * 10^{38} \sim 3.402823 * 10^{38}$ 内的数
双精度数	double	8	$-1.7977 * 10^{308} \sim 1.7977 * 10^{308}$ 内的数
长双精度	long double	8	$-1.7977 * 10^{308} \sim 1.7977 * 10^{308}$ 内的数
指针	< 一般类型关键字 > *	4	$0 \sim 2^{32}-1$ 内的整数

续表

类型	关键字	长度	值域范围
引用	<一般类型关键字>&		
数组	<一般类型关键字>[<N>]		
结构	struct<结构类型名>		
联合	union<联合类型名>		
类	class<类类型名>		
空值	void		

(4) 类型长度是指存储该类型值域范围内的任一个数据（又称为值）所占有的存储字节数，该字节数由系统规定，并且对任一数据都相同。如短整型长度为2，即存储每个短整数占用2个字节，对应16个二进制位；整型长度为4，即存储每个整数占用4个字节，对应32个二进制位；字符型长度为1，即存储每个字符占用1个字节，对应8个二进制位。

(5) 类型的值域范围是指该类型所对应的固定大小的存储空间按照相应的存储格式所能表示的值的范围。如对于有符号短整型来说，它对应2个字节的存储空间，存储格式为二进制整数补码格式，只能够表示（即存储）$-2^{15} \sim 2^{15}-1$，即 -32768 ~ +32767 之间的所有整数。若一个整数小于 -32768 或大于 32767，则它就不是该类型中的一个值，即它不是一个短整数。又如对于无符号字符类型来说，它对应1个字节的存储空间，存储格式为二进制整数无符号（隐含为正）格式，只能够表示 $0 \sim 2^8-1$，即 0 ~ 255 之间的所有整数。若一个整数小于0或大于255，则它就不是该类型中的一个值，即它不是一个无符号字符数据。

(6) 一个数的有效数字是指从该数最左边不为0的数字位起至最右边不为0的数字位止之间的每一个数字位，而这些数字位的个数称为该数的有效数字位数。如 3500，2.705，-0.278，63.00 和 0.00104 的有效数字位数分别为2，4，3，2 和 3。另外，若一个数带有指数部分，则它不影响整个数的有效数字位数。如 3.14，$3.14 * 10^5$，$314 * 10^{-6}$ 等都具有相同的有效数字的位数，即都为3位。

(7) 单精度型的值域范围是从 $-3.402823 * 10^{38}$ 至 $3.402823 * 10^{38}$ 之间的不超过7位有效数字的所有整数和小数。如 -372.65，$-0.14 * 10^{-6}$，0.0，+12.7，-6.45，100.0，$8.062 * 10^{25}$ 等都是单精度范围内的数。2.00708463 不是单精度范围内的一个数，若舍去它的最后两位有效数字，使之近似为2.007084，则就成为单精度范围内的一个数。

(8) 双精度型的值域范围比单精度型的值域范围更广，能够表示从 $-1.79769313486241 * 10^{308}$ 至 $1.79769313486241 * 10^{308}$ 之间的不超过15位有效数字的所有整数和小数。当一个数的有效数字的位数超过15时，则舍去第15位以后的所有位后，则可近似成为双精度范围内的一个数。在 VC++ 中长双精度型与双精度型定义完全相同。

(9) 在 VC++6.0 版本中，整型（int）和长整型（long ing）具有完全相同的长度和存储格式，所以它们是等同的。但在早期的 C++ 版本中，由于当时的机器字长为16位，所以整型和长整型的长度是不同的，前者为2个字节，后者为4个字节。无论如何，任一种 C++ 语言都遵循 short int 型的长度小于等于 int 型长度，同时 int 型长度又小于等于 long ing 型长度的规定。