

新概念

Visual C++ 6.0



完全模拟课堂培训的 22 讲多媒体教程

教程

王腾蛟
刘云峰 编著
汤晋主

北京科海集团公司 出品

北京科海培训中心

新概念 Visual C++ 6.0 教程

王腾蛟 刘云峰 汤晋立 编著

北京科海集团公司 出品

2001.5

内 容 提 要

本书由浅入深地讲解了使用Visual C++ 6.0 进行编程的方法和技巧,并提供了丰富可用的应用实例。

全书内容分为3篇:第1篇介绍面向对象的概念和C++语言的程序设计方法;第2篇着重讲述了利用Visual C++的集成开发环境设计应用程序的原理和过程,包括集成环境的安装、应用程序框架、AppWizard和ClassWizard的使用和结构原理、对话框和控件、绘图和打印等等;第3篇介绍了Visual C++的高级应用程序设计,包括Visual C++对数据库的访问、ActiveX控件和网络编程等等。

本书内容全面,涵盖面广,实例丰富,既是初学者很好的入门教程,又是程序设计人员必不可少的参考手册。

品 名:新概念 Visual C++ 6.0 教程

作 者:王腾蛟 刘云峰 汤晋立

责任编辑:刘 敏

出 品:北京科海集团公司

印 刷 者:北京门头沟胶印厂

发 行:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:28.125 字数:652 千字

版 次:2001 年 5 月第 1 版

印 数:0001~5000

盘 号:ISBN 7-89999-328-8

定 价:39.00 元 (1CD)

序 言

Visual C++是Microsoft公司开发的基于C/C++的可视化的集成开发工具，它也是Visual Studio中功能最为强大的、代码执行效率最高的开发工具。C/C++是应用范围非常广泛的程序设计语言，Windows操作系统本身的大部分代码就是用C/C++语言写成的，而Visual C++是使用C/C++语言编写Win32应用程序的集成开发环境，因而使用Visual C++来进行Windows应用程序的开发具有得天独厚的优势。学习并掌握Visual C++的编程方法，成为广大程序设计开发人员的迫切需要。

随Visual C++一起提供的Microsoft基本类库（Microsoft Foundation Class Library, MFC）为用户提供了Windows 95/NT环境下面向对象的程序开发界面，它为用户提供了大量预先编写好的类及其支持代码，用于处理标准的Windows编程任务，因此大大简化了Windows应用程序的编写工作。

Visual C++可以使用完全面向对象的方法来进行Windows 95/NT应用程序的开发，从而大大节省了应用程序的开发周期，降低了开发成本，也使得Windows程序员从大量的重复劳动中解救出来，相信读者会随着对Visual C++的逐步深入了解，而深切感受到这一点。

针对读者的不同需求，我们将全书内容分为3篇：第1篇重点介绍语言本身，其中包括面向对象的概念和C++语言的程序设计方法；第2篇是本书的核心，讲述了利用Visual C++的集成开发环境设计应用程序的一般原理和方法，包括集成环境的安装、应用程序框架、AppWizard和ClassWizard的使用和结构原理、对话框和控件、绘图和打印等等；第3篇介绍了Visual C++的高级应用程序设计，包括Visual C++对数据库的访问、ActiveX控件和网络编程等等。

参加本书编辑工作的还包括：

邓所云、童云海、曹新平、林斌、尹昊晖、洛向君、段紫辉、李红、王军、张凯、蒋丽华、胡建斌等，在此表示感谢。

本书编写过程中，得到了各方面的大力支持，在此表示诚挚的谢意。同时，由于作者水平有限，加上时间仓促，书中的缺点和错误在所难免，恳请广大读者批评指正。

作者

目 录

第1篇 C++语言基础

第1章 C++语言简介	1
1.1 什么是面向对象	1
1.2 C++对面向对象程序设计的支持	2
1.3 简单的C++程序实例	3
1.4 C++程序的编辑、编译和运行	5
1.5 调试程序错误	9
1.6 小结	10
1.7 练习题	10
第2章 基本数据类型与表达式	12
2.1 标识符	12
2.2 常量	13
2.3 变量	14
2.4 基本数据类型	15
2.5 运算符与表达式	17
2.6 小结	20
2.7 练习题	21
第3章 流程控制	23
3.1 条件语句	23
3.2 转向语句	26
3.3 循环语句	28
3.3.1 while语句	28
3.3.2 do-while语句	29
3.3.3 for语句	30
3.3.4 用if语句和goto语句构成的循环语句	33
3.3.5 循环嵌套	33
3.4 程序举例	35
3.5 小结	38
3.6 练习题	38

第4章 数组	39
4.1 一维数组	39
4.1.1 一维数组的定义	39
4.1.2 一维数组的引用	40
4.1.3 一维数组的程序举例	40
4.2 多维数组	43
4.2.1 多维数组的定义	43
4.2.2 多维数组的初始化	43
4.2.3 多维数组的引用	44
4.2.4 多维数组程序举例	45
4.3 字符数组与字符串的处理	48
4.3.1 字符数组的定义和初始化	48
4.3.2 字符数组的输入输出	48
4.3.3 字符串	50
4.4 字符数组程序举例	52
4.5 小结	55
4.6 练习题	55
第5章 函数	56
5.1 函数的定义	56
5.1.1 无参函数的定义	56
5.1.2 有参函数的定义	56
5.2 函数的调用	57
5.3 局部变量和全局变量	58
5.3.1 局部变量	59
5.3.2 全局变量	59
5.4 函数程序实例	60
5.5 小结	62
5.6 练习题	62
第6章 指针与链表	64
6.1 指针的基本概念	64
6.2 指针变量的定义	64
6.3 指针变量的引用	65
6.4 结构体	67
6.4.1 结构体的定义	67
6.4.2 结构体对象	68
6.4.3 结构体对象的引用	68
6.5 链表	69
6.5.1 链表的建立	69

6.5.2 链表的插入与排序	71
6.5.3 链表的删除	72
6.6 小结	74
6.7 练习题	75
第7章 类与对象	76
7.1 类的定义和对象的生成	76
7.2 成员函数和this指针	78
7.3 构造函数	79
7.4 析构函数	80
7.5 小结	81
7.6 练习题	81
第8章 继承和多态	82
8.1 类的继承	82
8.2 单一继承	83
8.3 多重继承	84
8.4 多态性和虚函数	86
8.4.1 运算符重载	86
8.4.2 虚函数	88
8.5 小结	90
8.6 练习题	91

第2篇 Visual C++初级程序设计

第9章 Visual C++集成开发环境	93
9.1 Visual C++工程及其文件构成	93
9.2 启动Visual C++	94
9.3 Visual C++集成开发环境综述	94
9.3.1 项目工作区	95
9.3.2 应用程序向导——AppWizard	97
9.3.3 类向导——ClassWizard	97
9.3.4 向导工具栏——WizardBar	97
9.3.5 Developer Studio中的快捷键和快捷菜单	98
9.4 小结	99
9.5 练习题	99
第10章 开发简单的Visual C++程序	101
10.1 AppWizard的使用	101

10.2 一个简单的应用程序	107
10.3 程序结构剖析	112
10.4 Win32编程基础	113
10.4.1 Win32数据类型	113
10.4.2 句柄	113
10.4.3 标识符命名	113
10.5 小结	114
10.6 练习题	115
第11章 应用程序框架与文档/视图结构	116
11.1 CPROGRAMApp类	116
11.2 CmainFrame类	120
11.3 CPROGRAMDoc类、CPROGRAMView类与文档/视图结构	123
11.4 CChildFrame类	130
11.5 应用程序框架中的文件列表	132
11.6 小结	132
11.7 练习题	133
第12章 Microsoft类库基础	134
12.1 Microsoft类库概述	134
12.2 根类: CObject	136
12.3 MFC应用程序框架结构类	136
12.4 MFC窗口类	139
12.5 MFC异常类	146
12.6 MFC文件类	147
12.7 绘图和打印类	147
12.8 ODBC类	150
12.9 Internet和网络类	150
12.10 OLE类	152
12.11 辅助类	153
12.12 小结	154
12.13 练习题	154
第13章 菜单、ClassWizard与消息映射	155
13.1 编辑菜单资源	155
13.1.1 系统生成的菜单	155
13.1.2 菜单的编辑	156
13.2 ClassWizard简介	157
13.3 建立消息映射	159
13.4 增强菜单项	164

13.5 小结	172
13.6 练习题	173
第14章 对话框	176
14.1 消息映射	176
14.2 定义对话框	179
14.3 为对话框增加功能	183
14.4 对话框的弹出	186
14.5 运行程序	188
14.6 程序分析	189
14.7 小结	196
14.8 练习题	197
第15章 控件	198
15.1 控件概述	198
15.2 静态控件	200
15.3 按钮控件	203
15.4 编辑框控件	207
15.5 列表框控件	210
15.6 组合框控件	215
15.7 滚动条控件	219
15.8 常用控件应用实例	221
15.8.1 对话框和控件的手工编辑	222
15.8.2 生成对话框类	225
15.8.3 为控件建立关联的成员变量	226
15.8.4 手工加入其他成员变量	229
15.8.5 成员变量的初始化	232
15.8.6 建立消息映射与响应函数	233
15.8.7 扩充响应函数	236
15.8.8 程序的完成	238
15.9 小结	247
15.10 练习题	247
第16章 实例分析	249
16.1 应用程序功能设计	249
16.2 编辑对话框	250
16.3 建立成员变量	253
16.4 建立菜单项	257
16.5 建立成员函数	261
16.6 小结	272

16.7 练习题	272
第17章 高级控件	274
17.1 高级控件简介	274
17.2 标签控件	274
17.3 树控件	291
17.4 旋转按钮控件	295
17.5 滑动条控件	297
17.6 进度条控件	300
17.7 小结	301
17.8 练习题	302
第18章 绘图和打印	303
18.1 绘图和打印类	303
18.2 绘图操作的实现过程	307
18.3 坐标与坐标模式	310
18.4 绘图与打印应用实例	314
18.5 小结	322
18.6 练习题	323

第3篇 Visual C++高级程序设计

第19章 访问数据库	325
19.1 MFC提供的数据库访问类	325
19.2 建立、连接数据源	327
19.3 建立访问数据库的应用程序	333
19.4 实现数据访问	340
19.5 增加和删除记录	342
19.6 程序分析	345
19.7 小结	352
19.8 练习题	352
第20章 ActiveX控件的设计与应用	353
20.1 ActiveX控件的概念	353
20.2 ActiveX控件容器	354
20.3 使用ActiveX控件	354
20.4 ActiveX控件的制作	359
20.5 小结	361
20.6 练习题	361

第21章 Socket编程与网络通信	362
21.1 Windows Sockets简介	362
21.2 CSocket 类中的成员函数.....	363
21.3 使用Windows Sockets编程.....	363
21.4 小结	422
21.5 练习题	423
第22章 动态链接库.....	424
22.1 概述	424
22.2 动态链接库(DLL)的结构	425
22.3 创建和使用动态链接库	426
22.4 本章小结	434
22.5 练习题	434

第1章 C++语言简介

C++是目前最流行的面向对象程序设计语言。它在C语言的基础上进行了改进和扩充，增加了面向对象程序设计的功能，更适合编制复杂的大型软件系统。这一章我们将引入面向对象的概念，并通过一个简单的C++程序来加深用户对面向对象程序设计方法的理解。

1.1 什么是面向对象

在面向对象的程序设计方法出现之前，占据主流的是结构化程序设计方法。对于复杂的问题，结构化程序设计采用模块化、自顶向下逐步求精的设计原则，因此结构化的程序往往清晰、易读。典型的结构化程序设计语言有C语言、PASCAL语言等，著名的UNIX操作系统的大部分代码就是用C语言编写的。

随着软件技术的发展，需要开发的系统越来越复杂。人们逐渐发现，对于大型软件系统来说，如果采用结构化的设计方法，设计、编程、测试和维护等工作都非常困难，而且有许多问题是结构化设计自身无法解决的。在这种背景下，产生了面向对象的方法，而面向对象的程序设计语言（Object-Oriented Programming Language，简称OOPL）也应运而生。

那么，什么是面向对象的方法呢？

面向对象方法的出发点和基本原则，就是使分析、设计和实现一个系统的方法尽可能地接近我们认识一个系统的方法。形象一点来说，就是使得描述问题的问题空间和解决问题的方法空间在结构上尽可能地一致。这样说可能太抽象，但随着我们对C++语言学习的深入，会逐步体会到这一点。

下面介绍面向对象方法的几个重要概念，作为后面学习C++语言的基础。

- **对象 (Object)**: 是由信息和对它进行处理的描述所组成的包，其结构如图1.1所示。

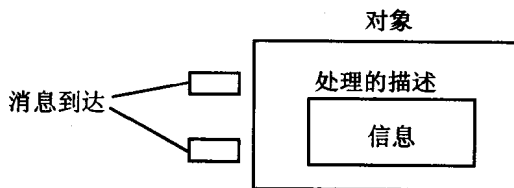


图 1.1 对象的图解

- 消息 (Message): 是对某个对象进行处理的说明。
- 方法 (Method): 是类似于过程的一个实体, 是对对象接受了某一消息后所采取的一系列操作的描述。
- 类 (Class): 是对具有共同特征的对象描述。
- 类的封装 (Encapsulation): 封装是把一类对象的状态 (用数据表示) 和方法 (用函数来表示) 封闭起来, 装入对象体中, 形成一个能动的实体。OOPL的封装机制模仿现实生活中的封装技术, 把一类对象的数据和函数封闭起来, 并提供访问它们的机制。外界只有调用对象的共有成员函数才能和对象交换信息, 这样就达到了封装的目的。
- 类的继承 (Inheritance): 是指新的类继承原有类的全部数据、函数和访问机制, 并可以增添新的数据、函数和访问机制。这样产生的新类叫子类或派生类, 原来的类叫父类或基类, 这种产生新类的方法叫类的派生, 也叫类的继承。如“汽车”是一个类, “轿车”就是它的子类, 而某辆实际的小轿车就是这个子类的一个对象。

面向对象的方法还有许多特征和概念, 如多态性、虚函数等, 我们会在后面具体介绍。

1.2 C++对面向对象程序设计的支持

C语言产生于1972年, 最早用来编写UNIX操作系统, 经过完善与改进后, 它发展成为一种通用的计算机语言。1983年进行标准化 (ANSI C) 后, 其发展更为迅速, 几乎所有操作系统都提供对C语言的支持。

1985年, AT&T贝尔实验室在C语言的基础上, 吸收了OOP的特点, 形成了面向对象的程序设计语言C++。

C++是一种灵活高效、可移植的面向对象程序设计语言。C++诞生以后, 发展极其迅速, 很多公司都研制了自己的C++版本, 并推出了多种C++的集成开发环境。

C++支持基本的面向对象的概念, 如对象、类、方法、消息、子类和继承性以及多态性等。表1.1给出了C++对这些概念的命名约定。

表 1.1 C++的概念及命名约定

面向对象的概念	C++的命名约定
对象	对象
类	类
方法	成员函数
实例变量	成员
消息	函数调用
子类	派生类/子类
继承	派生/继承

C++程序设计语言在C语言的基础上扩充了类、内联函数、运算符重载、变量类型、引用和自由存储管理运算符等语法。我们先通过简单的例子来了解C++语言的特点, 在下一

章再系统地学习C++语言的语法与编程方法。

1.3 简单的C++程序实例

C++是一个功能强大、方便灵活的程序开发语言，适合于编制复杂的大型软件系统。但若想熟练掌握它，则需要由浅入深、循序渐进地进行学习。

下面先给出两个简单的C++程序代码，然后仔细解释了每行代码的含义，用户可通过这两个例子初步领略C++的编程思想。

程序清单1.1：计算圆的面积和周长。

程序清单1.1 ex1-1.cpp

```
#include <iostream.h>
const double pi=3.14;
double calculate_area(int radius)
{
    return pi*radius*radius;
}
double calculate_perimeter(int radius)
{
    return 2*pi*radius;
}

void main()
{
    int input_radius;
    cout<<"请输入圆半径: ";
    cin>>input_radius;
    cout<<"圆面积为: "<<calculate_area(input_radius)<< "\n";
    cout<<"圆周长为: "<<calculate_perimeter(input_radius)<< "\n";
}
```

程序运行时显示：

请输入圆半径：

如果从键盘输入3并按回车键，程序会继续显示：

```
圆面积为：28.26
圆周长为：18.84
Press any key to continue
```

这时，程序运行已结束，按任意键可继续下面的操作。

程序中，如果首行有“//”的语句均为C++的注释行，程序执行时将会忽略注释行。

第2行的#include <iostream.h>是预处理包含指令，其中的iostream.h是C++提供的标准类库，它提供了输入输出流cin和cout以及输入输出运算符>>和<<的定义。

第3行说明了一个浮点型符号常量pi，该常量的固定值为3.14。

第4~7行是函数calculate_area的定义，该函数可以根据输入参数radius计算圆的面积。

第8~11行是函数calculate_perimeter的定义，它可以根据输入参数radius计算圆的周长。

第13~20行定义了主函数main()，其中main是函数名，它是程序执行的起点。

第16行是一个输出语句，`cout<<`表示从显示器输出后面的“请输入圆半径：”的内容。

第17行是一个输入语句，其中的`cin>>`表示从键盘输入一个整数给变量`input_radius`。

第18行是一个输出语句，先输出字符串“圆面积为：”，再输出由`calculate_area`计算出的半径为`input_radius`的圆的面积，最后输出“`\n`”，表示换行。

第19行也是一个输出语句，先输出字符串“圆周长为：”，再输出由`calculate_perimeter`计算出的半径为`input_radius`的圆的周长，最后输出换行符。

现在再解释`cin`和`cout`的用法：`cin`称作字符流输入，`cout`称作字符流输出，这两个函数都是在文件`iostream.h`中定义的。若想在程序中使用`cin`和`cout`，就必须用预处理包含指令`#include`将文件`iostream.h`嵌入到程序中。

运算符`<<`和`>>`的原意是表示算术左移和右移，在`iostream.h`文件中被重新定义（在面向对象中称作运算符重载），作为输入输出运算符使用。

`cout`中使用`<<`运算符，它将控制语句中的字符串或变量输出到标准设备上。`<<`运算符可以在一个语句中使用多次，比如：

```
cout<<"请输入姓名："<<endl;
```

该语句表示先输出字符串“请输入姓名：”，再输出`endl`。`endl`是换行符号，等效于“`\n`”。

`cin`中使用`>>`运算符，它可将键盘输入的信息存于变量中，当遇到空格便立即停止输入。一个`cin`语句中可以多次使用`>>`符号，比如：

```
cin>>a1>>a2;
```

注意：当使用`cin`输入语句时，与标准C语言的输入语句不同，它输入信息的变量可以是任意类型的，而且不要求带有地址指示符`&`。

程序清单1.2：使用类封装计算圆的面积和周长。

程序清单1.2 ex1-2.cpp

```
#include <iostream.h>
const float pi=3.14;
class Circle
{
public:
    int radius;
    double calculate_area()
    {
        return pi*radius*radius;
    }
    double calculate_perimeter()
    {
        return 2*pi*radius;
    }
};
void main()
{
    Circle theCircle;
    cout<<"请输入圆半径：";
    cin>>theCircle.radius;
    cout<<"圆面积为："<<theCircle.calculate_area()<<"\n";
```

```
cout<<"圆周长为: "<<theCircle.calculate_perimeter()<<"\n";  
}
```

该程序的运行结果与程序1.1完全相同,但是程序结构却大不一样。程序的4~16行定义了一个类 Circle,其中有公共类型的变量 radius 和两个函数 calculate_area() 和 calculate_perimeter()。主程序中的第1行生成了一个类Circle 的对象theCircle,然后我们将输入的半径值存入该对象的radius变量,再利用该对象的两个函数分别计算圆面积和周长。

可能用户还有不清楚的地方,因为这个例子涉及了C++中OOPL的一些基本概念,这些概念在后面还要详细讨论。

1.4 C++程序的编辑、编译和运行

这一节,我们将初步学习使用Microsoft Visual C++ 6.0来编辑、编译和运行简单的C++程序。只有自己动手调试程序,才能深刻地理解C++程序。在这里,我们仅用到Microsoft Visual C++ 6.0中很简单的功能,在下一篇中,再深入介绍Microsoft Visual C++ 6.0的各项功能及其使用方法。

1. 程序的编辑

- 启动Microsoft Visual C++ 6.0: 按照图1.2所示,单击“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”,进入如图1.3所示的Visual C++ 6.0集成开发环境。

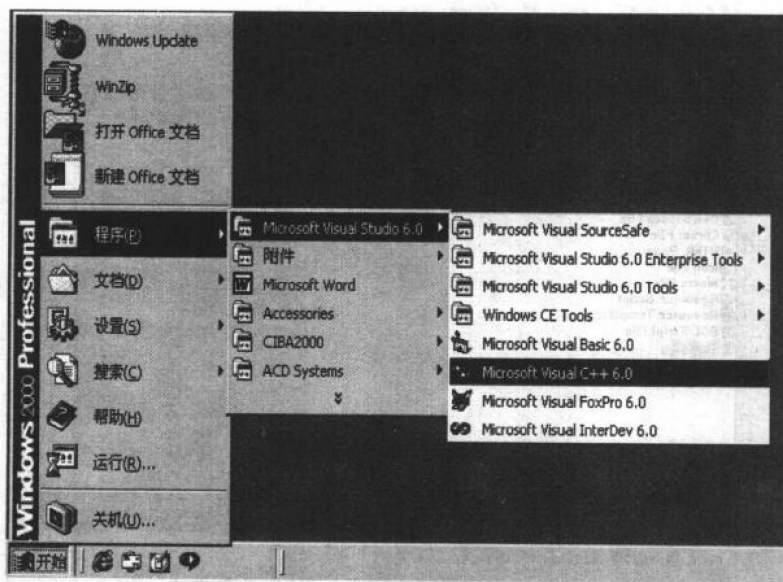


图 1.2 由“开始”菜单进入 Microsoft Visual C++ 6.0 集成开发环境

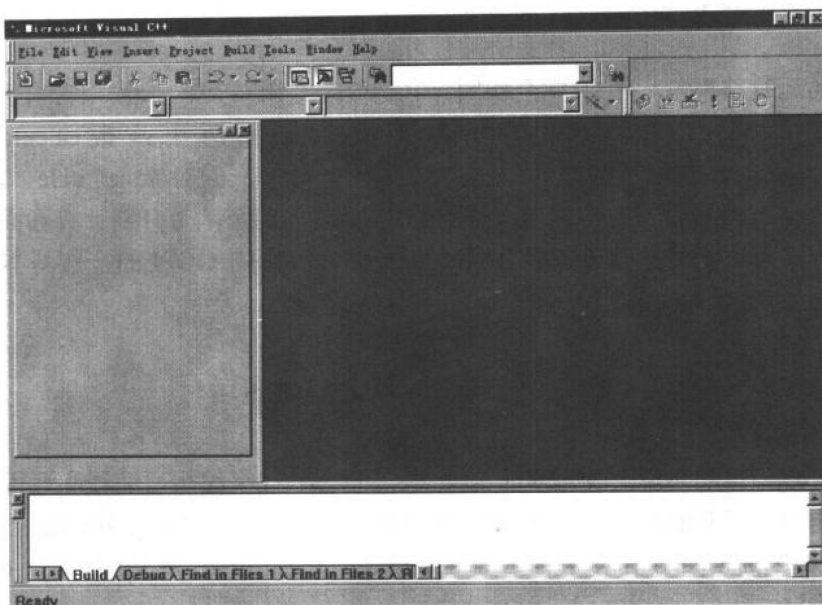


图 1.3 Microsoft Visual C++ 6.0 集成开发环境

- 创建文件：单击主菜单中的File→New，再单击New对话框中的Files标签。在Files选项卡左半部分列出的选项中，单击最后一项Text File，并在右半部分创建文件的名称和位置，如图1.4所示。最后单击OK按钮。

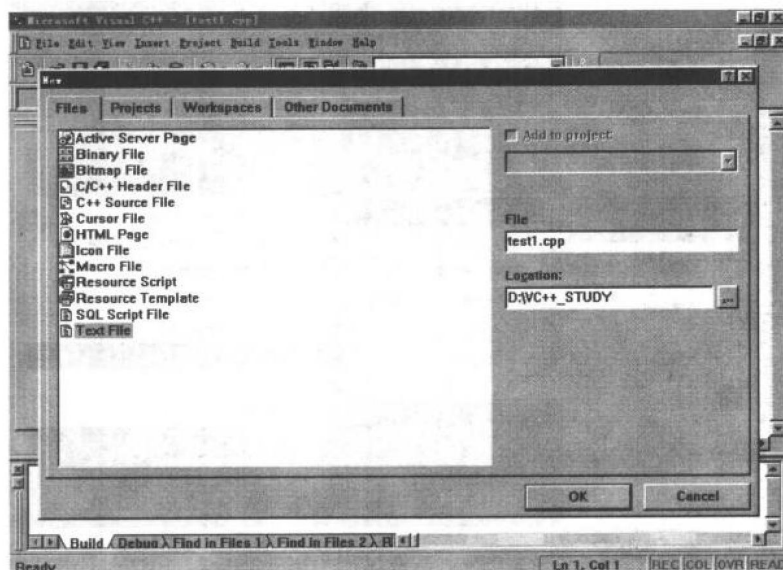


图 1.4 创建新的文件

注意：文件的扩展名一定要写成.cpp，cpp是英语c plus plus（即C++）的缩写，以.cpp为扩展名的文件都是C++源程序。