

普通高等教育“十五”国家级规划教材

高等职业教育技能型紧缺人才培养试用

程序设计基础

——逻辑编程及 C++ 实现

陆虹 主编

陶霖 周晓云 副主编

高等教育出版社

内容提要

本书是普通高等教育“十五”国家级规划教材,主要内容包括程序设计概述、数据类型与数据的输入输出、表达式语句、控制语句、数组、指针与文件、结构和函数等。

本书针对高职高专学生的认知特点,以及在开始学习程序设计时的实际情况,使用“示范实例—模仿实例—实训任务”的体例。每章首先给出学习目标,明确对知识、技能掌握程度的要求,然后选用大量实用、趣味及学生身边所熟悉的问题,作为实例与实训任务,便于教师组织互动式的教学。

本书不仅注重编程能力的培养,而且强调程序调试能力的培养,通过调试实例提高学生解决问题的能力,增强编程的信心。

本书适用于高等职业学校、高等专科学校、成人高校、示范性软件职业技术学院、本科院校举办的二级职业技术学院、继续教育学院和民办高校使用,也可供技能型紧缺人才培养使用,还可供计算机专业人员使用。

图书在版编目(CIP)数据

程序设计基础——逻辑编程及 C++ 实现/陆虹主编.

北京:高等教育出版社,2003.8

ISBN 7-04-012643-5

I. 程... II. 陆... III. ①程序设计—理论—高等学校—教材②C语言—程序设计—高等学校—教材
IV. ①TP311.1②TP312

中国版本图书馆 CIP 数据核字(2003)第 063390 号

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 010-82028899

购书热线 010-64054588

免费咨询 800-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷

开 本 787×1092 1/16

印 张 22.25

字 数 540 000

版 次 年 月第 1 版

印 次 年 月第 次印刷

定 价 27.80 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

出版说明

为加强高职高专教育的教材建设工作,2000年教育部高等教育司颁发了《关于加强高职高专教育教材建设的若干意见》(教高司[2000]19号),提出了“力争经过5年的努力,编写、出版500本左右高职高专教育规划教材”的目标,并将高职高专教育规划教材的建设工作分为两步实施:先用2至3年时间,在继承原有教材建设成果的基础上,充分汲取近年来高职高专院校在探索培养高等技术应用性专门人才和教材建设方面取得的成功经验,解决好高职高专教育教材的有无问题;然后,再用2至3年的时间,在实施《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》立项研究的基础上,推出一批特色鲜明的高质量的高职高专教育教材。根据这一精神,有关院校和出版社从2000年秋季开始,积极组织编写和出版了一批“教育部高职高专规划教材”。这些高职高专规划教材是依据1999年教育部组织制定的《高职高专教育基础课程教学基本要求》(草案)和《高职高专教育专业人才培养目标及规格》(草案)编写的,随着这些教材的陆续出版,基本上解决了高职高专教材的有无问题,完成了教育部高职高专规划教材建设工作的第一步。

2002年教育部确定了普通高等教育“十五”国家级教材规划选题,将高职高专教育规划教材纳入其中。“十五”国家级规划教材的建设将以“实施精品战略,抓好重点规划”为指导方针,重点抓好公共基础课、专业基础课和专业主干课教材的建设,特别要注意选择一部分原来基础较好的优秀教材进行修订,使其逐步形成精品教材;同时还要扩大教材品种,实现教材系列配套,并处理好教材的统一性与多样化、基本教材与辅助教材、文字教材与软件教材的关系,在此基础上形成特色鲜明、一纲多本、优化配套的高职高专教育教材体系。

普通高等教育“十五”国家级规划教材(高职高专教育)适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

教育部高等教育司
2002年11月30日

前 言

Visual C++ 已是被广泛应用的程序开发平台,越来越多的学校采用它作为程序设计课程的教学平台。

根据这一实际需求,我们以 VC++6.0 为平台,编写程序设计基础的系列教材,包括《程序设计基础——逻辑编程及 C++ 实现》、《程序设计基础——面向对象及 C++ 实现》和《程序设计基础——可视化及 VC++ 实现》。

本书针对高职高专院校学生的认知特点,本着从具体到抽象的原则,从具体技术问题引出知识要点,使用“示范实例—模仿实例—实训任务”的体例。每章首先给出学习目标,明确对知识、技能掌握程度的要求,以求学生熟练掌握技能,并在此基础上理解相关知识。同时便于教师组织互动式教学。

笔者根据在长期教学实践与研究中积累的经验,考虑学生在开始学习程序设计时所面临的种种困难,不过分追求知识、技能的“完整性”和“系统性”。而从实际应用的需求出发,大胆舍弃了一批繁杂的细节,提炼、整合了程序设计基础中最基本、最核心的内容,引导学生集中注意力掌握主干知识与技能。

从实用性、趣味性、学生身边所熟悉的问题出发,选择大量的实例与实训任务,以求提高学生的学习兴趣。

我们在每一章开始,以“学习目标”的形式,明确对知识、技能掌握程度的要求,在知识和技能二者之间,尤其强调技能。

对一些技巧性强或容易模糊的技术要点,我们采用了“思考与验证”的形式,启发并引导学生把理性思考和动手验证结合起来。

在实际应用中,程序的调试具有极其重要的作用与地位,因此,我们从编程的实际需求出发,不仅培养学生的程序编写能力,而且强调培养学生的程序调试能力。为了追求这一目标,本系列教材精心设计了调试实例,以求学生在一个一个“挫折”中,提高解决问题的能力,增强编程的信心。

本书由上海第二工业大学陆虹、陶霖、周晓云编写,其中,第 6 章由周晓云编写,其余各章由陆虹、陶霖合作编写。

衷心感谢华东师范大学黄国兴教授对本书所给予的指导与帮助。

限于编者水平,书中难免存在疏漏和不足之处,恳请广大师生及读者批评指正。

编 者

2003 年 7 月

目 录

第 1 章 程序设计概述.....	1	3.3 编写简单程序.....	56
1.1 程序设计入门.....	1	3.4 程序执行过程的跟踪.....	59
1.1.1 算法和程序.....	1	3.5 实训任务.....	65
1.1.2 一个简单的 C++ 程序.....	3	第 4 章 控制语句.....	67
1.1.3 C++ 程序的结构.....	4	4.1 if 语句.....	67
1.2 创建和运行一个 C++ 程序.....	5	4.2 while 语句和 for 语句.....	75
1.3 程序设计的基本流程.....	12	4.3 switch 语句与 break 语句.....	92
1.4 VC++ 的集成开发环境.....	13	4.4 程序调试中断点的使用.....	101
1.4.1 基本概念.....	13	4.5 实训任务.....	109
1.4.2 主窗口.....	14	第 5 章 数组.....	111
1.4.3 主要菜单功能.....	15	5.1 数组.....	111
1.5 实训任务.....	16	5.2 一维数组.....	112
第 2 章 数据类型与数据的输入输出.....	17	5.2.1 声明一维数组.....	112
2.1 数据在计算机中的存储.....	17	5.2.2 访问数组的元素.....	112
2.2 数据在 C++ 程序中的表示.....	19	5.2.3 初始化数组元素.....	113
2.2.1 变量.....	19	5.2.4 初始化字符数组.....	117
2.2.2 常量.....	23	5.2.5 一维数组的应用.....	121
2.3 数据的输入与输出.....	29	5.3 二维数组.....	134
2.3.1 数据输出.....	29	5.3.1 声明二维数组.....	134
2.3.2 数据输入.....	31	5.3.2 访问与初始化二维数组.....	135
2.4 实训任务.....	35	5.3.3 二维数组的应用.....	138
第 3 章 表达式与语句.....	36	5.4 实训任务.....	146
3.1 表达式.....	36	第 6 章 指针与文件.....	149
3.1.1 算术表达式.....	37	6.1 指针变量.....	149
3.1.2 赋值表达式.....	42	6.1.1 变量的地址.....	149
3.1.3 关系表达式.....	45	6.1.2 指针变量的声明.....	151
3.1.4 逻辑表达式.....	47	6.1.3 通过指针变量访问数据.....	153
3.1.5 条件表达式.....	50	6.1.4 指针变量和数组.....	154
3.1.6 强制类型转换运算符.....	51	6.1.5 指针变量和字符串.....	159
3.1.7 逗号表达式.....	54	6.1.6 内存的动态分配.....	161
3.2 语句.....	55	6.2 文件操作.....	167
3.2.1 表达式语句.....	55	6.2.1 文件变量的声明.....	167
3.2.2 复合语句.....	56	6.2.2 文件的打开与关闭.....	167
3.2.3 空语句.....	56	6.2.3 文件的格式化读写.....	169

6.3 实训任务.....	178	8.1 函数的基本概念	212
第7章 结构	180	8.2 函数的定义与调用	217
7.1 结构类型.....	180	8.2.1 函数定义	217
7.1.1 定义结构类型	180	8.2.2 函数调用	218
7.1.2 声明结构类型变量	182	8.2.3 函数声明	218
7.1.3 访问结构变量成员	184	8.2.4 参数传递	220
7.1.4 访问结构变量	184	8.2.5 函数返回	222
7.1.5 初始化结构成员	186	8.2.6 局部变量与全局变量	223
7.2 结构数组.....	188	8.3 应用举例.....	228
7.2.1 声明结构数组	188	8.4 函数的跟踪调试	255
7.2.2 访问结构数组元素	189	8.5 综合实例.....	259
7.2.3 初始化结构数组	190	8.6 实训任务.....	273
7.3 应用实例.....	194	参考答案	279
7.4 实训任务.....	210		
第8章 函数	212		

第 1 章 程序设计概述

学习目标

- 了解算法和计算机程序的概念
- 了解算法的自然语言表达方式
- 了解简单 C++ 程序结构
- 了解程序设计的基本步骤
- 掌握 Visual C++ 6.0 集成开发环境下程序的编辑、编译、连接和执行操作

1.1 程序设计入门

1.1.1 算法和程序

“能为我计算两个数的平均值吗？”

“要计算的是哪两个数？”

“3 和 5”

“加起来，除以 2”

“结果是多少？”

“4”

“请再算两个数”

“哪两个数？”

“23763965432.2187563 和 8456233445446456.43701”

“加起来，除以 2”

“结果是多少？”

“你让计算机算去！”

被提问的小学生虽然没能立即算出结果，但他知道两件事。第一，他知道正确的方法和步骤，那就是：

- (1) 获得要计算的数；
- (2) 求出两个数之和；
- (3) 再除以 2；
- (4) 报告计算结果。

第二，他知道，复杂的计算可以借助计算机完成。

计算机具有比人强大不知多少倍的计算能力,但是,计算机的“智力”其实连这位小学生都不如,因为它只会机械地执行最简单的基本运算,必须依靠人为它安排好方法和步骤,才能通过一步一步的运算,最终完成复杂任务。

今天,这位小学生已经成长为大学生,这些年来,计算机技术也在飞速发展,但是,计算机与他相比的劣势却依然未变。通过“程序设计基础”这门课程的课堂学习和上机实训,他将能够针对要完成的计算任务,编排出正确的方法和步骤,并且用计算机能够接受的形式,把方法和步骤告诉计算机,指挥计算机完成任务。

我们通常把解决问题的方法和步骤称为“算法”。算法可以用许多种不同的形式来表达,像上面那样以(1)、(2)、(3)、(4)逐条列出,是一种形式,这种自然语言的形式能让人理解,却不能让当今的计算机理解,要让计算机理解和执行,就必须使用专门的语言。解决问题的方法和步骤,以计算机能够理解的语言表达出来,就被称为“程序”,这样的专门语言则被称为“程序设计语言”,C、C++、C#、BASIC、Java等,都是程序设计语言。C++是目前得到广泛使用的一种。用C++语言表达的程序,称为C++程序。

自然语言是给具有灵性的人看的,程序设计语言是要给死板的计算机“看”的,因此,前者往往不那么精确,语法、格式可以比较自由,后者则必须符合一套严格的语法规则,一个字母或标点符号的错、漏、增,都会导致计算机无法接受或者错误“理解”,对此,我们在学习和实践中必须充分重视,努力适应,直到熟悉并掌握。

【示范实例】

用自然语言表达解决下面问题的算法:

计算一位学生3门课程的考试总分和平均分。

- (1) 获得要计算的3个数;
- (2) 求出3个数之和;
- (3) 把和除以3;
- (4) 报告两项计算结果。

【示范实例】

用自然语言表达求解一元二次方程的算法。

- (1) 获得一元二次方程 $ax^2 + bx + c = 0$ 的3个系数 a 、 b 、 c ;
- (2) 计算 $b^2 - 4ac$,得到中间结果 d ;
- (3) 计算 d 的算术平方根 s ;
- (4) 分别计算 $(-b + s)/(2a)$ 和 $(-b - s)/(2a)$;
- (5) 报告两个计算结果。

这个实例采用了代数符号来表示数据和运算,使叙述变得简洁、精确。以后我们会看到,程序设计语言中以相似的形式来表达数据和运算。

【模仿实例】

用自然语言表达求解下面应用题的算法：

两车相距若干距离，分别以各自的速度相向行驶，计算它们需要多少时间才能相遇。

- (1) 获得两车的初始距离 s (km)、两车的速度 v_1 和 v_2 (km/h)；
- (2) 计算 $v_1 + v_2$ ，得到中间结果 v ；
- (3) 计算 s/v ；
- (4) 报告计算结果。

1.1.2 一个简单的 C++ 程序

下面通过一个简单的程序实例，来体会一下 C++ 程序。

```

/*
    源文件名 :ch1_01.cpp
    功能 :求两数之和
*/
#include <iostream.h >

void main( )
{
    int a ,b ,sum ;

    cout << "请输入两个整数：" ;           //提示输入
    cin >> a >> b ;                          //输入两个数
    sum = a + b ;                             //计算
    cout << "两数之和是：" << sum << endl ; //输出计算结果
}

```

这个程序的功能是求两个整数之和，计算机执行后，在屏幕上显示如下内容：

```

请输入两个整数 7 5
两数之和是 :12

```

其中的“7 5”带有下划线，表示这是从键盘输入的。

下面是对这个程序的一些解释，读者目前只需在表面上了解，暂且不必深究，随着以后的学习，会逐渐加深理解。

(1)“/* */”是注释，是写给人看的说明文字，目的是帮助有关人员了解、管理程序。计算机在接受程序时，会自动忽略从“/*”到“*/”的内容。

以“//”开头直到行结束的部分也是注释，如“//提示输入”、“//输入两个数”等，与上述注释的区别是它的范围只到本行的结束为止，不能跨行。这种注释称为行注释，而前一种注释则称为块注释。两种注释都可以出现在程序中的任何位置。

注释的文字可以用英文，也可以用中文，前者便于键盘输入，但对英文水平有一定要求，编写

时可以根据实际情况随意选择。

程序中的空行是为了视觉上的清晰而加的,像注释一样,不影响程序的功能。出于同样的目的,程序中还可以加入空格、制表符,但不能打断语句的基本成分,例如“cout”的中间就不能加空格。至于哪些东西算基本成分,在以后的学习中会逐渐体会。

(2)“#include <iostream.h>”是预处理命令。

(3)“void main(){...}”是程序的主体。“main()”表示主函数;“main”是它的函数名。由一组大括号“{ }”括起来的是函数体。

(4)函数体中的“int a b sum;”是变量声明,通知计算机为将要用到的变量a、b、sum安排存储空间。

(5)函数体中还包含完成功能所需的一系列语句,相当于自然语言表达中的步骤(1)、(2)、(3)……,这些语句就是要求计算机执行的命令。

在这个例子中有4句语句:

“cout << “请输入两个整数 :”;”的作用是在屏幕上显示提示文字,要提示的内容“请输入两个整数 :”跟在运算符“<<”(连续两个小于号)的后面。

“cin >> a >> b;”的作用是从键盘输入两个数值给变量a和b,两个变量分别跟在运算符“>>”(连续两个大于号)的后面。

“sum = a + b;”的作用是使变量sum的值成为a与b之和。

“cout << “两数之和是 :” << sum << endl;”的作用是依次向屏幕输出每个运算符“<<”后面的内容,先是文字“两数之和是 :”,再是变量sum的值。

1.1.3 C++ 程序的结构

C++ 程序的一般结构可以用下面的示意图表示:

名称			上例中的内容	
注释区			/* 文件 :ch1_01.cpp 功能 :求两数之和 */	
声明区			#include <iostream.h >	
程序区	主 函 数	函数首部		void main()
		函 数 体	{	{
			声明部分	int a b sum ;
			执行部分	cout << “请输入两个整数 :” ; cin >> a >> b ; sum = a + b cout << “两数之和是 :” << sum << endl ;
		}	}	
	其他函数			

位于开头的注释区是对整个程序的说明,其他各部分中都可以包含注释,对程序局部内容加以说明。

在程序区之前的是声明区,它通常包含若干行预处理命令,以“`#include`”开始的预处理命令是最常用的。

程序区中包含着函数,一个程序中一定有且仅有一个主函数,另外根据需要,还可以有若干其他函数。每个函数都有函数首部和函数体,有自己的函数名。主函数的函数名一定是 `main`。

函数体以“`{`”开始,以“`}`”结束,其中包含变量声明和若干语句(执行部分),一个语句可以按一定规则分成多行,而一行也可以写多个语句。每个语句以分号结束。

在 C++ 程序中,同一字母的大小写被当作两个不同字母,例如 `sum` 和 `Sum` 不能混为一谈。

程序编写完成后,需要以文件形式保存到磁盘等介质,供计算机进一步处理,也便于长期保存和修改。以“.cpp”为文件名后缀的文件就保存着 C++ 程序。用任何文本编辑工具打开这种文件,都可以查看、修改程序的内容。上面程序注释区中的“文件 `xh1_01.cpp`”就是说明文件名的。

C++ 程序中通常都有以“`#include`”开始的预处理命令,命令中包含着以“.h”为后缀的文件名,文件名写在“`<`”和“`>`”之间或一对“`"`”之间,表示程序需要使用这个文件。这种文件称为头文件。头文件的内容其实也是 C++ 程序的组成部分,只不过这部分内容不需要我们自己编写,通过预处理命令就能加到我们的程序中。在本章的后面,还将进一步解释有关的细节。

1.2 创建和运行一个 C++ 程序

在这一节中,我们以上一节介绍的程序为例,动手经历一下 C++ 程序设计的实际过程。下面的步骤所需要的软件环境是 Microsoft Visual C++ 6.0 (简称 VC++) ,这是支持 C++ 程序设计的专用工具软件。

在此过程中,VC++ 将把我们的小程序称为“工程”(Project),就像对待真实的应用程序一样。

(1) 在 Windows 环境中,选择菜单命令“开始”|“程序”|“Microsoft Visual Studio 6.0”|“Microsoft Visual C++ 6.0”,启动 VC++ ,出现如图 1-1 所示的窗口。

(2) 选择命令“File”|“New”,在弹出的“new”对话框中,在“Project”标签下选择“Win32 Console Application”,在“Location:”文本框中

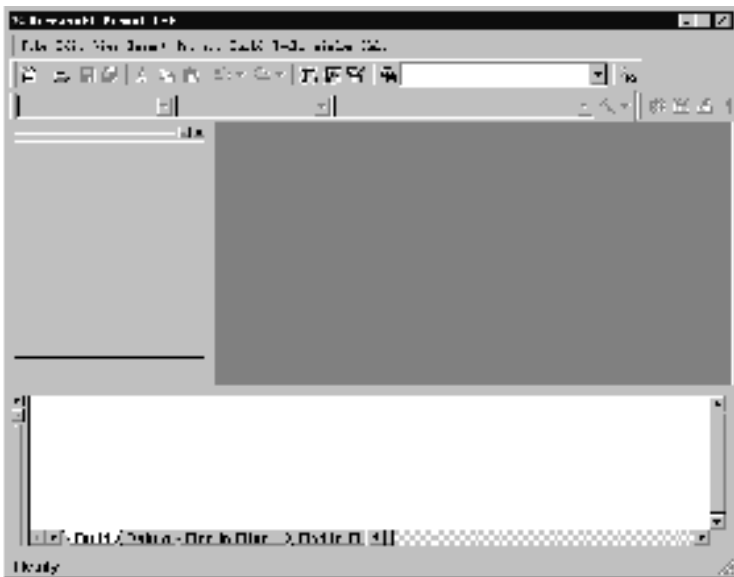


图 1-1

指定新建工程的路径 c:\student ,再在“ Project name :”文本框中输入新建工程的名称 ch1_01 ,如图1-2所示。

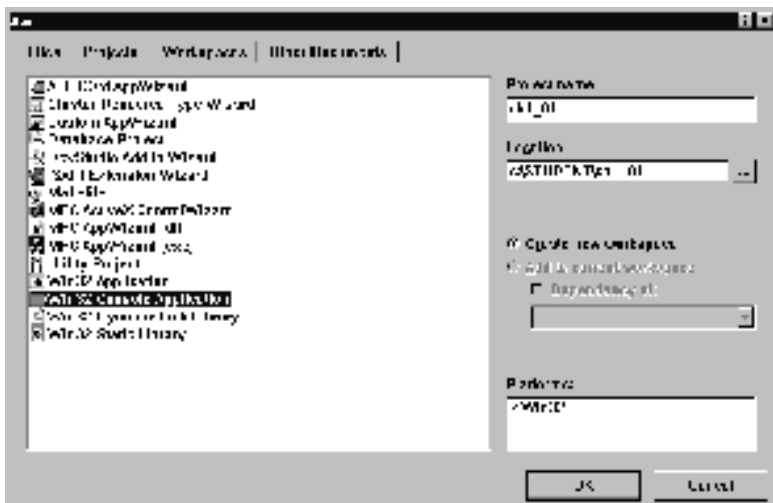


图 1-2

(3) 单击“ OK ”按钮 ,显示对话框如图 1-3 所示。



图 1-3

(4) 选中“ An empty project ”后 ,单击“ Finish ”按钮 ,在弹出的“ New Project Information ”对话框中 ,单击“ OK ”按钮 ,出现如图 1-4 所示的窗口。

(5) 选择命令“ Project |> Add to Project |> New...” ,出现如图 1-5 所示的对话框。

(6) 在“ Files ”选项卡下 ,选择“ C++ Source File ” ,在“ File ”文本框中输入“ ch1_01 ” ,如图 1-6 所示。

(7) 单击“ OK ”按钮 ,出现如图 1-7 所示的窗口 ,右边的“ ch1_01.cpp ”框中有字符输入光

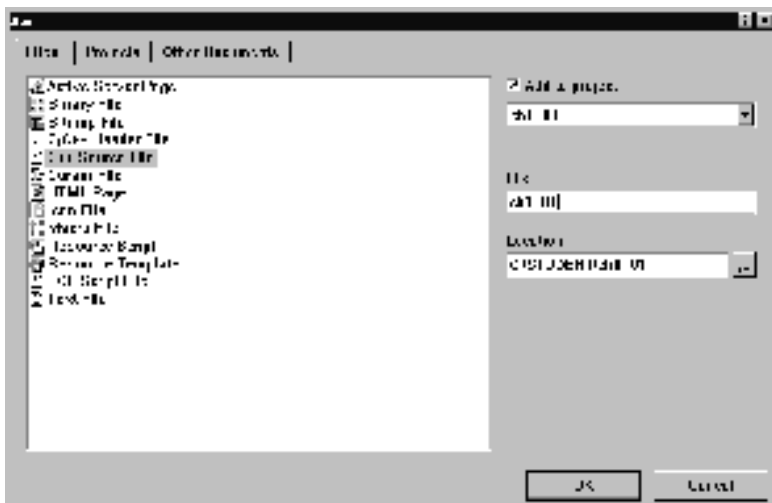


图 1-6

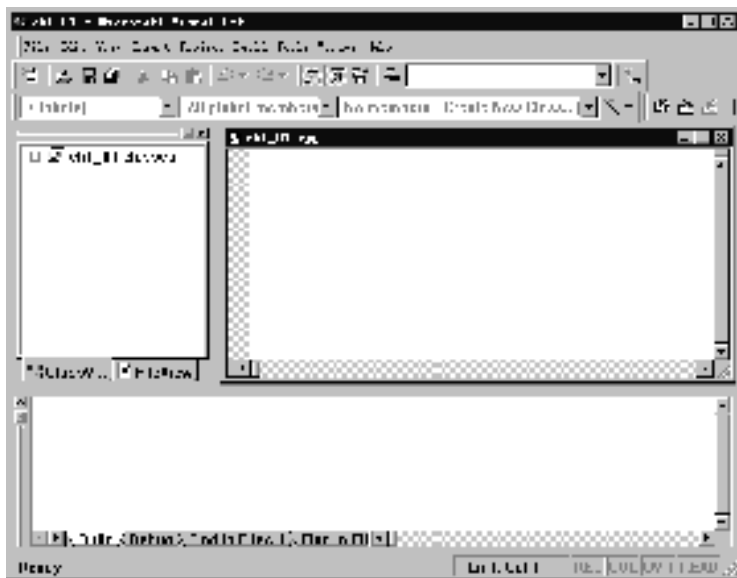


图 1-7

(9) 选择命令“Build”|“Compile ch1_01.cpp”结果应当如图 1-9 所示。

窗口下部的框内最后一行说明在程序中发现了多少错误,如果不是“0 error(s), 0 warning(s)”,则要检查输入的程序,纠正错误,再重复这一步,直到没有错误为止。

如果程序经过了修改,再次选择菜单命令“File”|“Save”,把经过修正的程序保存到文件 c:\student\ch1_01.cpp,取代原来有错误的程序。

(10) 选择命令“Build”|“Build ch1_01.exe”结果如图 1-10 所示。

(11) 选择命令“Build”|“Execute ch1_01.exe”,出现如图 1-11 所示的运行窗口。

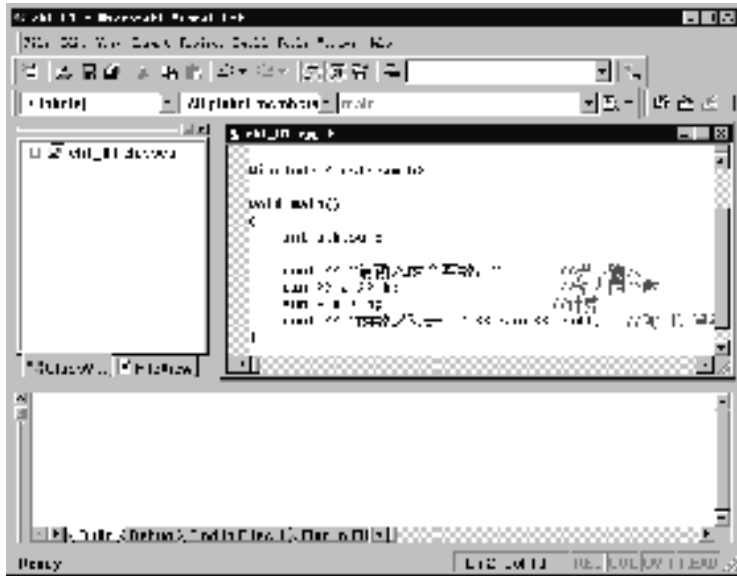


图 1-8

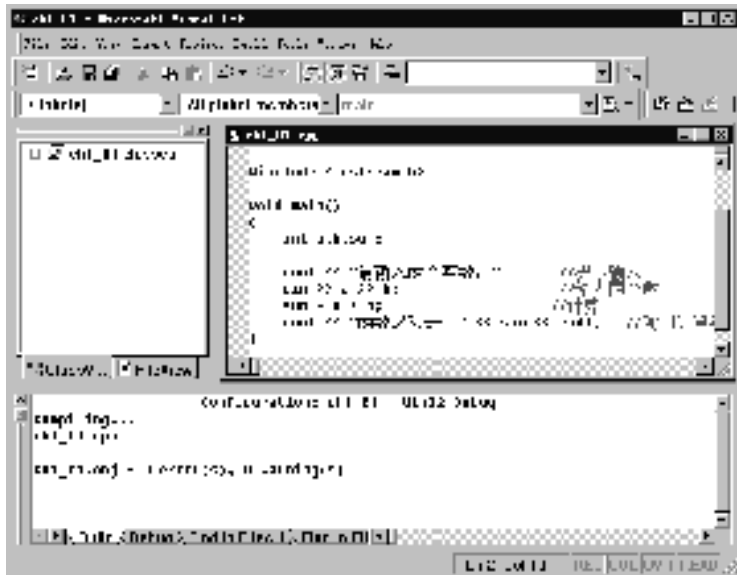


图 1-9

从键盘输入 7 和 5 ,中间用空格隔开 ,回车。屏幕显示程序的运行结果如图 1-12 所示。观察运行结果后 ,按任意键 ,运行窗口消失。

这个程序在一个黑色的窗口中运行 ,远不像常见的应用程序运行时那样色彩丰富、形象生动 ,原因是在上面的步骤(2)中选择了“ Win32 Console Application ”(Win32 控制台应用程序) ,这是一种最简单的程序 ,适合于初学者打好基础。在这本书的整个学习过程中 ,我们编的

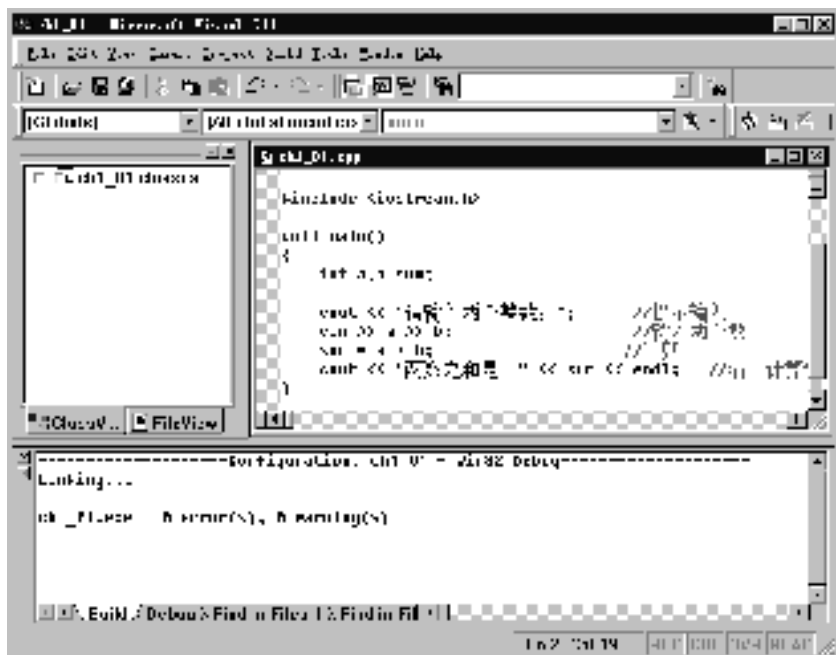


图 1-10

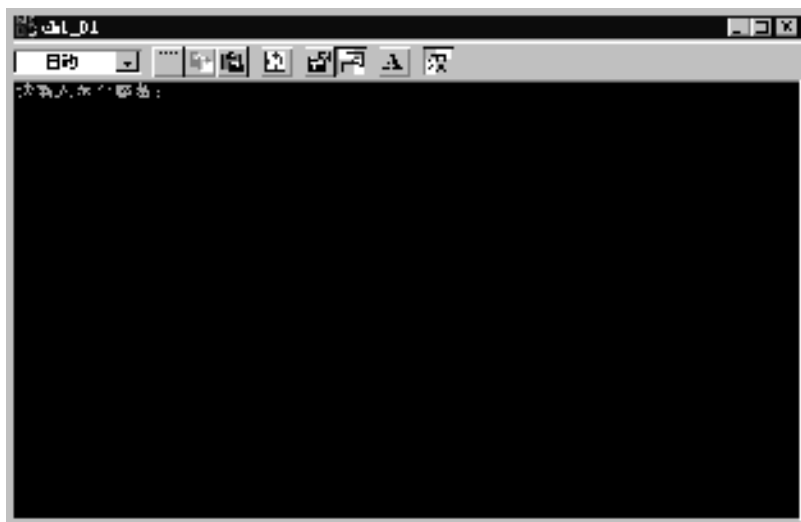


图 1-11

程序都是这样的面貌。在掌握了程序设计的基本概念和方法后,我们将通过后继课程的学习,编出漂亮的程序来。

(12) 选择命令“File”|“Exit”,关闭 VC++。

在掌握了上面的操作方法后,请读者再继续进行下面的各步操作,这是为后面的进一步学习

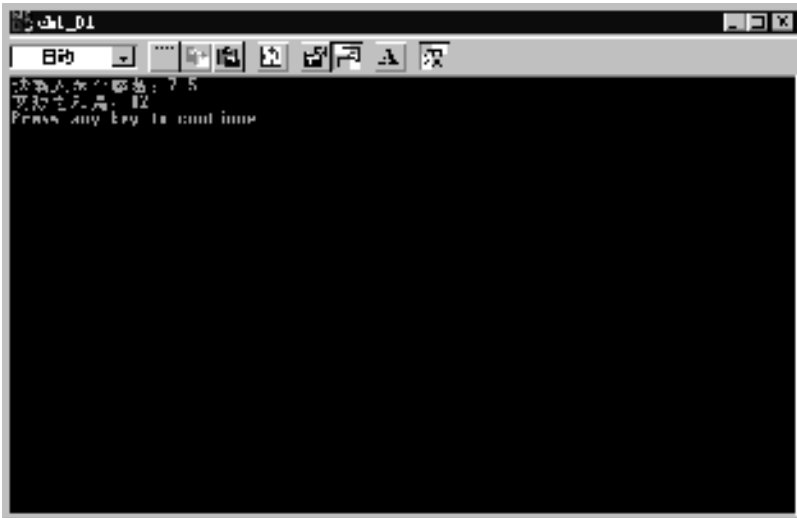


图 1-12

做准备。

(1) 重复上面的操作,但不关闭 VC++。

(2) 在程序中把一个“cout”中的字母 o 改成数字 0,再做第 9 步,观察并记录结果后纠正。

(3) 在程序中删除一个分号,再做第 9 步,观察并记录结果后纠正。

(4) 把程序中“int a, b, sum;”中的一个西文逗号改成中文逗号,再做第 9 步,观察并记录结果后恢复。

(5) 把程序中“cout << “请输入两个整数:”;”中的一个中文冒号改成西文冒号,再做第 9 步,观察并记录结果后恢复。

(6) 在程序中删除花括号“{”,再做第 9 步,观察并记录结果后恢复。

(7) 在程序中删除预处理命令,再做第 9 步,观察并记录结果后恢复。

(8) 在程序中把一个“//”改成“/”,再做第 9 步,观察并记录结果后恢复。

(9) 在程序中把“求两数之和”改成“求两数之差”,再做第 9 步到第 11 步,观察并记录结果后恢复。

(10) 在程序中把“sum = a + b”改成“sum = a - b”,再做第 9 步到第 11 步,观察并记录结果后恢复。

(11) 在程序中把“int a, b, sum;”改成“int a, b, c, sum;”,再做第 9 步,观察结果后,不加纠正,继续做到第 11 步,观察并记录结果后恢复。

(12) 制造你感兴趣的其他差错,观察并记录结果后恢复。

(13) 归纳出你的结论:怎样的差错会在第 9 步被发现?怎样的差错不影响程序的运行结果?怎样的差错不在第 9 步被发现,却影响程序的运行结果?

(14) 关闭 VC++。