

21 世纪计算机系列教材

程 序 设 计 语 言

——VC++编程方法

朱志辉 主编

夏宽理 主审

上 海 交 通 大 学 出 版 社

内 容 简 介

本书从 C++语言最基本的概念入手，由浅入深，综合大量的编程实例，引导初学者从入门到掌握 C++语言的程序设计方法，每一章都有学习目标和大量的习题，帮助读者掌握相关知识。主要包括 VC++语言基本数据类型、控制结构、函数、数组、指针、类、面向对象特性、流文件及实践练习。

本书的特点是通俗易懂、面向应用、重视实践，以任务驱动式介绍 C++语言的编程方法。教材中的例子都配有相应解释或注释语句，方便读者阅读理解。本书还附有光盘，包括本书教案及全部例题的源文件，可供自学者使用。

本书内容丰富，不仅适合作为计算机专业程序设计课程的入门教材，还适合作为其他各类计算机教育的 C++语言程序设计课程的教材，也可供广大学习 C++语言程序设计的技术人员参考。

目 录

1	程序设计初步	1
1.1	计算机的组成和工作原理	1
1.2	程序和程序设计	2
1.3	数据与数据表示	4
1.4	过程程序设计、模块程序设计和面向对象程序设计	4
1.5	运行第一个 C++ 程序	5
2	C++ 程序的组成	12
2.1	C++ 的 cout 和 cin	12
2.2	基本数据类型	15
2.3	变量与常量	19
2.4	一个简单的 C++ 程序	26
3	程序的结构流程	34
3.1	C++ 的简单语句	34
3.2	C++ 的顺序结构	35
3.3	C++ 的 if 分支结构	36
3.4	switch 多选择分支	40
3.5	C++ 的 while 循环结构	45
3.6	for 循环结构	47
3.7	do-while 循环结构	50
3.8	转向语句和多重循环	51
3.9	应用实例	53
4	函 数	67
4.1	函数的建立与调用	67
4.2	函数参数的传递	72
4.3	全局变量、局部变量和静态变量	79
4.4	重载函数和递归函数	83
4.5	应用实例	87
5	多文件结构	102
5.1	工程文件	102
5.2	头文件	103
5.3	编译预处理	104
6	数组与字符串	107
6.1	一维数组定义和引用	107
6.2	一维数组的初始化	109
6.3	一维数组应用举例	110
6.4	二维数组	114

6.5	字符串	118
6.6	数组应用：模拟	121
6.7	向函数传递数组	122
7	指 针	130
7.1	指针的定义	130
7.2	指针与指针所指的对象	131
7.3	指针的引用和运算	134
7.4	字符指针与二级指针	137
7.5	指针与数组	140
7.6	指针与函数	144
7.7	动态内存分配	151
8	类	168
8.1	结构	168
8.2	类	175
8.3	构造函数和析构函数	185
9	面向对象的特性	197
9.1	面向对象的特性	197
9.2	类的继承	197
9.3	重载多态性：运算符函数	208
9.4	动态多态性：虚函数	213
9.5	友元	219
9.6	模板	226
9.7	综合应用	233
10	文 件	242
10.1	I/O 流	242
10.2	使用 I/O 成员函数	243
10.3	文件的输入与输出	246
11	C++应用技巧	250
11.1	Windows 环境下的编程	250
11.2	Windows 的编程实例	253

前 言

本书是程序设计基础的入门教材，选用 VC++ 作为语言工具。本书充分体现高校计算机专业的培养目标，在编写过程中突出实践性教学环节，特别是 C++ 程序设计方法的基本技能训练，强调培养学生与实践中的应用能力，进而激发学生的创新能力，推进素质教育。编者严格按照教育部的大纲要求进行编写，内容安排充分体现 21 世纪高等教育教材的特色，能够按照学生学习程序设计方法的心理顺序整合知识，按模块组织教学，使用时可以根据需要灵活选择，弹性较大。

在教材的编写中加强了改革意识，汲取国内外教材编写的先进思想，编写体系和内容选取注重培养学生适应信息化社会要求和编程应用的需要。通过合理组织课程内容，以当前比较流行的应用软件 Visual C++ 作为程序设计的平台，使学生掌握 C++ 的程序结构、面向对象、指针与类等的基本知识和基本技能，能够解决日常生活及工作中的常见问题，在此基础上提高分析问题和解决问题的能力和获取计算机新知识、新技术的能力，在毕业后具备较强的实践能力、创新能力和创业能力。

从实际出发，加强教材的实用性和适用性，注意到专业与应用结合的同时，使教材适用于不同学校、不同学习程度的学生使用。本教材具有如下特点：

落实教学大纲精神。根据教育部面向 21 世纪的教材要求，本教材充分体现了实用性和创新性，突出了 C++ 的程序设计方法，而不是程序设计语言的语法，让学生在实践中掌握编程方法的同时熟悉 C++ 的有关语法，以达到触类旁通的目的。内容安排充分体现新知识、新技术的特点。

加强能力培养。教材体现“能力本位”的指导思想，注重培养学生的实践应用能力、创新能力和自学能力，坚持学以致用原则，为学生就业和继续学习打下基础。

强调实践性教学环节。注意使计算机的专业学习与应用相结合，通过计算机的应用案例及模仿操作，引导学生进行思考、分析、研究和讨论，使学生领会计算机在日常工作、生活中的应用，提高实践中的应用能力。每章配有大量的习题和实践练习。

使学生适应今后的变化和发展。教材符合学生的认知水平，内容的选择和组织注意横向与纵向的整合、逻辑顺序与心理顺序的整合、灵活性与创造性的整合等，尤其重视共性和迁移性，使学生学会举一反三、灵活运用，适应今后的变化和发展。

配备辅助教与学的光盘。其内容包括每一章的知识点、习题、实践练习等，并提供一些学生探究活动的案例、教与学的素材等。

本书是为初学者编写，随着计算机技术的发展，面向对象程序设计语言应该作为基础课的教学语言。本书的第 1 章介绍学习程序设计所必需的一些基础知识，第 2~7 章介绍基于面向对象介绍了程序设计的基本方法，第 8~10 章介绍基于面向对象介绍了程序设计的思想和实例，第 11 章介绍 VC++ 的 MFC 类库和 Windows API 的功能。对于本科学生，建议第 1 章到第 7 章安排 54 理论学时，第 8 章到第 11 章安排 36 理论学时，上机学时至少 11，并适当安排一些习题课。

本书的第 1~5、10、11 章由朱志辉编写，第 6、7 章和第 8、9 章分别由穆玲玲和咎红英编写，书中所有程序在 Visual C++6.0 下运行通过。全书由朱志辉主编，复旦大学夏宽理教授主审。

由于编写时间仓促，水平有限，错误之处在所难免，恳请读者批评指正。

编 者
2002 年 5 月

1 程序设计初步

学习程序设计的目的是利用计算机解决实际问题，主要学习内容包括分析问题、解决问题的方法，以及程序设计语言。其中程序设计语言是问题求解方法的描述工具。因此，考虑后续学习数据结构等课程和将来工作的需要，本书选用 C++ 作为程序设计的教学语言。

本章学习目标：

- 了解计算机的组成和工作原理，区别什么是硬件和软件
- 懂得什么是计算机程序，能够写出解决问题的步骤，知道什么是算法
- 认识程序设计的发展过程，懂得什么是高级语言
- C++ 程序设计语言的特点，初步理解对源程序编辑、编译、连接和执行的过程

1.1 计算机的组成和工作原理

1.1.1 计算机的组成

完全不懂得计算机的组成也可以编写计算机程序，但是知道计算机的工作过程，则能够使我们更有效地理解和编写计算机程序。所以我们在介绍程序设计语言之前简单回顾一下计算机的组成。

大多数计算机的组成如图 1.1 所示，它由 CPU、内存单元、输入和输出设备组成。内存单元相当于有序的存储仓库，它分为很多小块，每一小块都编了号码，通常称为存储单元地址。计算机在这些内存单元里存放程序指令和有关数据，然后通过存储单元地址来找出所需要的指令或数据。我们可以通过输入设备（如键盘）或辅助存储设备（如硬盘）把计算机程序或数据放入内存单元，再通过 CPU 控制和处理后，由输出设备（如显示器）或辅助存储设备把处理结果反馈给用户。这就是计算机处理问题的简单工作过程。那么，如何让计算机按人的意愿去处理问题呢？这就需要通过编写计算机程序来完成。

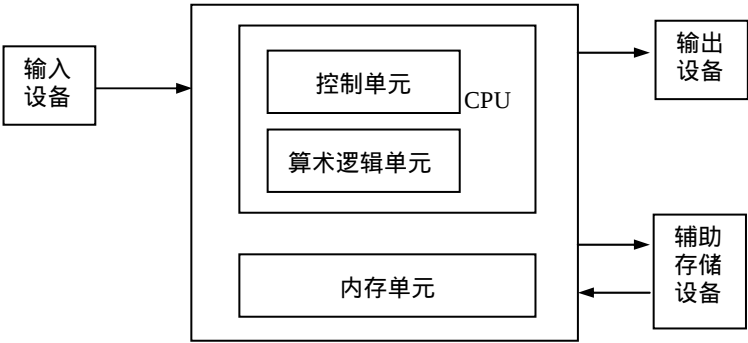


图 1.1 计算机的基本组成

硬件——计算机的物理组成部分，称为计算机的硬件部分；即看得见、摸得着的部件。

软件——描述问题、设计开发过程的各类文档和指挥计算机按人的意愿进行工作的指令集合，称为计算机的软件部分，其中指挥计算机工作的指令集合又称为程序。

1.1.2 高级程序设计语言

计算机是一种有“思维”的机器，所以，指挥计算机的指令必须能够让这种机器接受。不管是哪一种计算机，都有自己能识别的基本指令系统，这些指令集合称为机器语言。使用机器语言又难写又难读，因此人们就用一些助记符来表示这些机器指令，这种比机器语言更容易读写的语言称为汇编语言。

无论机器语言或汇编语言都与人类的语言有较大差别，而且，每一种计算机的指令系统都有差别，对于解决同一问题所编写的程序也有所不同；为了解决这一问题，人们就设计出一种接近人类自然语言的代码来编写处理问题的程序，把这种代码语言称为高级程序设计语言（简称高级语言）。但是计算机并不能直接理解高级语言，因此需要把高级语言编写的程序转换为计算机能理解的机器语言程序，实现这一功能的指令集合称为编译程序或解释程序。一种高级语言普及后，在各种计算机上都建立这种语言的编译程序或解释程序，这样就能使同一种高级语言所编写的程序被编译成不同的机器语言程序，以便能在各种计算机上运行。C++就是一种高级语言。

1.2 程序和程序设计

在日常生活和工作中，人们都用一些逻辑字符序列来表示处理问题的过程，它表达了人们的习惯和思维。例如，在计算器处理数学、语文、英语三科成绩的平均分，其过程是：

输入：数学成绩 X ，语文成绩 Y ，英语成绩 Z ；

处理： $S = X + Y + Z$ ；

输出：平均分 $= S / 3$ ；

实际上，我们按照上述的步骤就可以计算出任意三个数的平均值，这种用来解决问题的有限多个步骤组成的具体过程通常称为算法。因此，算法是由一些能够机械执行的操作步骤组成的，我们再把这此操作序列转换为计算机能识别的字符序列，那么就可以利用计算机来解决问题。所以，如果我们把上述具体问题看作是一个实体（Entity），那么程序就是实体在计算机中的体现，是表示一些操作序列的计算机指令的集合；程序设计就是把问题世界转换为程序世界的过程，或者说是为解决某一具体问题而编写计算机程序的活动。如图 1.2 所示。

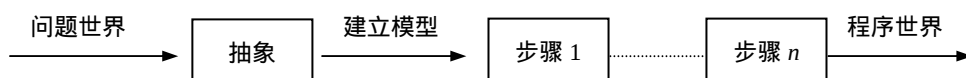


图 1.2 程序设计

所以，要编写一个程序，必须掌握一种程序设计语言和它的开发环境，同时要熟悉问题世界的知识和掌握把问题世界转换为程序世界的方法（通常称为程序设计方法）。

C++是一种优秀的程序设计语言，它包括了C的全部特征、属性和优点，同时添加了支持面向对象编程（OOP）的设施，比C语言更容易学习。C++语言包括过程性语言部分和类部分，过程性语言部分与C语言并无本质的差别，只是功能增强了，而类部分是C语言所没有的。通常在学习C++时，可以先学习过程性语言部分，再学类部分。从共有的过程性语言部分这个意义上说，学习C++语言，无须先学习C语言。（如果读者已熟悉C语言，可以从第4章开始学习，前3章很大程度上是一种复习）。

开发一个C++程序，如图1.2所示，通过对问题的分析和抽象后，建立问题的模型，再把模型转换为有限的步骤序列，然后按照C++的语法规则组成C++的语句序列即C++的源程序。把源程序从键盘输入到计算机内，以文件形式保存成文件后缀为.CPP的过程称为编辑，编辑好的源文件要把它编译成后缀为.OBJ的目标文件后，再连接成后缀为.EXE的可执行文件，运行可执行文件便可以得到处理结果。如图1.3所示。

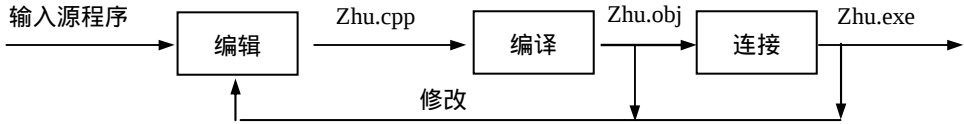


图 1.3 C++程序开发过程

C++程序开发过程：创建源代码文件(.CPP)
编译成目标文件（.OBJ）
连接生成可执行文件（.EXE）

使用C++语言进行程序设计时，一定要了解支持C++语言的是哪一版本，因为C++语言尚有多个标准，不同的版本是有一些区别的。目前，C++有以下标准：ISO（国际标准化组织）标准，NCITS（国家信息技术化委员会）标准，ANSI（美国国家标准协会）标准，其中ANSI标准使用最广，所以本书的代码是使用ANSI标准代码。而且不同的C++语言开发环境中，完成图1.3的过程也会有差异。本书是以VC++6.0版本为程序的开发和运行环境。

Java语言是一种适合于分布式计算的新型面向对象程序设计语言，它可以看作是C++语言的派生语言，从C++语言中继承了大量的语言成分，但抛弃了C++语言中容易引起问题的功能，增加了多线程、网络程序设计等方面的支持，掌握了C++语言就很容易学会Java语言。

？ 小结

所谓程序，是用计算机能接受的特定语言对所要处理的数据以及处理的方法和步骤所作的完整而准确的描述；该描述过程称为程序设计。对所要处理的数据的描述就是指明数据采用何种类型的数据结构；对处理方法和步骤的描述也称为算法。

文本编辑器——生成的文件是纯文本文件，没有如何格式化命令的字处理器，例如WINDOWS的记事本、C++自带的文本编辑器等。C++的源程序必须用文本编辑器建立和修改。

编译器——把高级语言编写的源程序转换为机器代码的处理器。源程序被编译后生成目标文件，还必须用连接器把目标文件转变为可执行文件。C++都自带编译器和连接器。

1.3 数据与数据表示

在问题世界中，用来描述我们所认识或感觉到的信息或者是用来表示处理问题过程的符号序列都称为数据。计算机程序中的数据一般分为字符型和数值型。字符型数据包括字母、数字、标点等符号，如果所处理问题的数据不需要运算，我们都把它作为字符型数据。在程序中又把单个字符定为字符型(char)，把多个字符定为字符串型(string)。数值型数据包括整数、实数(在计算机中的表示又称为浮点数)、有符号数和无符号数等，如果要进行运算的数字数据，我们都把它作为数值型数据。在程序中又可分为整型(int或integer)、浮点型(float, 又称单精度型)、双精度型(double)等。

无论是字符型或数值型数据，在计算机内部都用二进制编码来表示，二进制编码是利用二进制数制只含“0”和“1”两个数码的特点，通过这两个数码的不同编码方式，可以分别用来表示各种数值、字符、声音、图形、图像、动画等数据。今天，使用最广的是ASCII编码方式，它是针对字符的一种编码方案，ASCII码规定每个字符由7位二进制数表示，共可定义128种符号。数据使用ASCII编码的文件称为ASCII文件。在计算机系统中，通常用8位的二进制数表示一个字符，它和ASCII码的定义一样，只是在最左边加了一个0。例如，在ASCII码中，字符“a”用1100001表示，而在计算机中则用01100001表示(它相当于十进制的97，参阅附录1.3 ASCII码表)。

由于计算机内的所有数据(包括数值型和字符型等)都以二进制形式(0或1的编码)表示，所以二进制的“位”是数据的最小单位，用比特(bit)表示，一个比特记为1bit。通常把8位的二进制数编为一组叫做一个字节(byte)，计算机中存储数据都以字节作为基本存储单元。

存储容量单位及换算公式：

1 个字节记为 1B，1B = 8bit	
1 千字节	1KB=1024B=2 ¹⁰ 字节
1 兆字节	1MB=1024KB=2 ²⁰ 字节
1 千兆字节	1GB=1024MB=2 ³⁰ 字节

1.4 过程程序设计、模块程序设计和面向对象程序设计

1.4.1 过程程序设计

过去，人们把程序看作是处理数据的一系列过程(procedure)，一个过程或函数(function)是指一组特定的一个接一个顺序执行的指令，它们完成某一专门任务或计算出一个结果值。数据与过程是分离的，程序设计的技巧主要是在处理过程之间的调用及完成任务或计算的算法上，关心的是数据的变化，过程程序设计的关键是程序设计语言所提供的子程序及参数传递形式。从程序组织的角度来看，子程序是用于有组织地管理大量算法的有力工具。

为解决过程程序设计可能存在的问题，结构化程序设计应运而生。结构化程序设计的主要思想是把功能分解并逐步求精，在问题世界中一个任务太大或十分复杂时，就把它分解为若干个小任务，在程序世界中使一个程序分解为若干个过程组成，每一过程完成一个确定的小任务。由于过程程序设计是根据数据的不同来决定程序的编写，所以，人们把程序描述为：

程序=算法+数据结构（包含数据和数据类型）

即算法和数据结构分别是一个独立整体，两者分开设计，以算法（过程或函数）为主。

1.4.2 模块程序设计

随着软件的发展，考虑到程序的可重复使用，人们越来越注重于系统整体关系和对数据的组织，他们把数据与操作数据的相关过程称为模块（module），程序是根据模块的需要来划分，并使数据隐藏在模块中，程序设计的技巧主要是模块中过程的设计，在模块中既包含具体问题的数据，又包含这些数据上的操作。由于模块程序设计是把算法和数据结构（数据和数据类型）看作一个独立功能模块，所以，程序就被描述为：

程序=（算法+数据结构）

即算法与数据结构是一个整体，算法总是离不开数据与数据类型，算法含有对数据的访问，算法只能适用于特定的数据结构。

1.4.3 面向对象程序设计

我们现在使用计算机的方式是利用按钮、菜单或窗口，通过选择后触发计算机去执行某一事件动作，这种编程方法称为事件驱动，是一种交互性很强的编程方法。面向对象程序设计可以满足上述的需求，它能实现软件组件的可重用性，并把数据和操作数据的过程结合起来作为一个整体——对象，程序设计的技巧是以类作为构造程序的基本单位，它具有封装、抽象、继承、多态性等特点。由于在面向对象中，算法与数据结构被结合在一起成为一个类，问题世界本身就是一个对象世界，任何对象都具有一定的属性与操作，所以，程序又被描述为：

对象 = （算法 + 数据结构）

程序 = （对象 + 对象 +）

一个对象是对一个客观实体的数据和方法的封装体。

1.5 运行第一个 C++ 程序

学习一门程序设计语言的最好方法是首先运行一个简单的程序，从中体会程序的结构。这一节给出一个小示例程序，它可以帮助初学者了解如何编写 C++ 程序，以及如何输入、编译和运行一个 C++ 程序。我们知道，过去在没有计算机的时候，很多问题的解决都是靠人类的聪明才智，有了计算机，仍然离不开人，只是为解决问题的方法、时间、空间提供更大的方便，如何利用计算机程序帮助我们去解决问题呢？我们回顾 1.2 节的一个简单例子：

【例 1.1】假设数学成绩是 82，语文是 89，英语 90，计算三科成绩的平均分。

// 一个完整的 C++ 源程序 zhu1_1.cpp

```

/* 本程序的作用是根据问题的要求：计算三个数的平均值，把处理结果在屏幕上显示出来；
*/
#include <iostream.h>           // 预处理行

int main()
{   int score1, score2, score3;    // 定义表示数学、语文、英语成绩的变量
    int average;                  // 定义表示平均分的变量

    score1=82;                    // 把数学成绩记录在 score1 中
    score2=89;
    score3=90;
    average = (score1 + score2 + score3) / 3; // 计算出平均分，把结果记录在 average
                                           // 由于运算符“/”两边都是整型数，所以
                                           // 运算结果只取整数部分，C++称为整数除

    cout << "The average score is " << average << endl; // 显示出处理的结果
    return 0;
}

```

我们把上述程序输入并执行（不用输入中文的内容），屏幕上显示的结果应该是：

The average score is 87

在 Visual C++ 6.0 环境下运行一个新程序的上机操作步骤：

打开 VC++窗口，单击菜单【File】→【New】；

在弹出窗口中选择标签“Projects”（工程），选中“Win32 Console Application”项，然后在“Project name”框中输入工程名，如 zhu1，在“Location”框中输入保存源文件的路径名（文件夹名），单击【OK】按钮；

在弹出窗口的选项中选择“An Empty Project”，单击【Finish】按钮，在下一窗口再单击【OK】按钮；

再从 VC++菜单条中单击菜单【File】→【New】；

在弹出窗口中选择标签“Files”（文件），选中“C++ Source File”，并在“File”框中输入文件名，如 zhu1_1；（注意：此时，“Add to project”框应该是刚才输入的工程名，“Location”框应该是刚才输入的路径名）；

在编辑窗口中输入源程序，保证源程序正确后，单击菜单【File】→【Save】存盘；

在 VC++菜单条中单击【Build】→【Compile】，检查程序无编辑错误后单击【Build】→【Build】，检查无连接错误后，再单击【Build】→【Execute】执行此程序，运行结束按任意键退回 VC++窗口；

注意：如果再输入另一个新程序时，单击菜单【File】→【Close Workspace】，然后再重复上述步骤。

注意：C++程序的大、小写字母是被看成不同的两个字符。

我们暂时不必完全理解上述 C++ 程序的所有内容，只是通过这个简单的例子掌握如何运行一个 C++ 程序，以及了解 C++ 程序的基本结构。

程序 zhu1_1.cpp 实现了例 1.1 提出的问题，从这个简单的程序例子可以看出，任何一种高级语言的程序都如写文章一样，具有一定的格式。在这个 C++ 程序中，以 “//” 开头直至该行行尾的内容属于注释(comment)部分，用于说明或解释程序的功能，它对程序的运行效果不产生任何影响，计算机运行程序时会跳过所有的注释；如果注释内容不止一行，可以用 “/*” 和 “*/” 把注释内容括起来（如上例的第二行注释）。

C++ 的程序都从主函数 main() 开始逐行执行每一语句，每一行语句都用 “;”（分号）作为结束标志，一个语句可以写在多个程序行上，一个程序行也可以写多个语句；程序的所有语句用一对花括号 “{” 和 “}” 括起来，每个程序必须有而且只能有一个主函数 main()，否则编译时会产生错误。另外，我们注意到在主函数 main() 前面有一行以符号 “#” 开头，这是预处理的标志，C++ 每次启动编译器时，首先运行以 “#” 开头的预处理行，include 是一条预处理的命令，其含义是：后面跟的是用一对尖括号 “<” 和 “>” 括起来的一个文件名（称为头文件），编译器在编译程序之前，先把该文件读入到程序中，相当于把该头文件的内容粘贴到程序中。上述程序中用到的 “cout” 语句（实质是一个输出流）的作用是向屏幕输出，它已在 iostream.h（输入输出流，见第 10 章介绍）文件中定义，所以，在程序中要使用 cout 输出流就必须在主函数 main() 之前插入 “#include <iostream.h>” 预处理命令行，它使预处理器在保存编译器的所有 .h 文件的目录下查找 iostream.h 文件。从上面分析可知，一个 C++ 程序的基本结构是由编译预处理和主函数 main() 组成，建议在必要的地方加入注释，以增强程序的可读性。注意：main() 的前面加上 “int”，和主函数的最后一行加上 “return 0;”，是为了保证 main() 函数返回一个整型，建议初学者养成这个习惯，其作用以后再介绍。

cout 的用法：键入 cout，接着键入输出重定向运算符 “<<”（连续两个小于符号 <），后面紧跟的内容全部输出到屏幕显示，如果是输出一串字符，一定要用双引号 (") 括起来；endl 的作用是使输出光标转到下一行（即 “换行”）。

编译错误的处理：出现编译错误的原因很多，通常是由于输入错误或其他不注意的问题引起，一般编译器能够提示在哪一行发生什么错误。一般出现的错误有 “error” 和 “warn” 两类，前者使程序不能运行，必须保证 “0 error” 才能使程序运行；后者不影响程序的正常运行，但建议养成一个好习惯，必须也保证是 “0 warn” 才运行程序。例如出现如下错误：

```
zhu1_1.cpp
D:\C++教材\zhu\zhu1_1.cpp(10) : error C2146: syntax error : missing ';' before
                                identifier 'score3'
D:\C++教材\zhu\zhu1_1.cpp(15) : warning C4508: 'main': function should return a
                                value; 'void' return type assumed
Error executing zhu1_1.exe.

zhu1_1.obj - 1 error(s), 1 warning(s)
```

上述信息表示出现一个错误和一个警告，它指出在 zhu1_1.cpp 程序中的第 10 行（cpp(10)表示在源程序的第 10 行）有句法错误，在 “ score3 ” 前一行语句缺少分号 “ ; ”。在源程序的第 15 行出现警告错误，指+明返回 int 型值的 main()函数必须有 return 语句返回一个值。当保证编译和连接时都是 “ 0 error(s) 0 warning(s) ” 时才执行该程序。

你可以故意在程序中制造一个错误来看看。

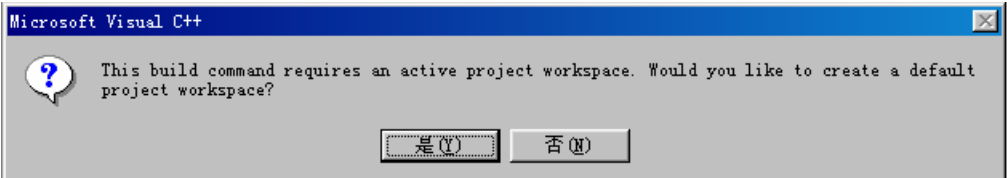
一个 C++ 程序只能有一个主函数 main()。

如果已经存在一个.CPP 的 C++源程序，也可以用下面步骤打开它：

打开 VC++窗口，单击菜单【File】→【Open】；

在弹出对话框中找到该文件所在的文件夹，选中该文件名（例如 zhu1_1.cpp），单击【打开】按钮；则可把该文件调入 VC++的编辑窗口；

在 VC++菜单条中单击【Build】→【Compile】，弹出对话框如图所示。单击【是】按钮；



检查程序无编辑错误后单击【Build】→【Build】，检查无连接错误后，再单击【Build】→【Execute】执行此程序。运行结束按任意键退回 VC++窗口。

注意：运行程序后，会生成很多中间文件，初学者只要保留 C++源程序.cpp 文件，其他都可以删除，以免占用空间，下次运行时系统会自动再生成。

说明：其中*.dsp 是 C++的项目文件，*.dsw 是工程工作区文件，*.opt 是工程工作区配置文件（见第 11 章）。

本章小结

本章介绍了计算机的基本知识和程序设计语言的基本功能，不论使用那一种程序设计语言，一个程序都是客观实体在计算机中的实现，程序设计就是一个建模过程，或者说是把特定问题领域的概念转换为程序设计语言可表达的概念。用 C++语言编写的程序是由一个主函数 main()和若干个函数或类组成，它必须经过编辑、编译、链接过程才可以运行。

附录 1.1 程序设计语言的组成

每一种程序设计语言都有规定的词汇，词汇集由标识符、保留字、特殊字符、数等组成。当我们学习每一种程序设计语言时，应该注意它的语法和语义。

语法：表示语言的各个构成记号之间的组合规则。

语义：表示的含义。

程序设计语言很多，但它们的组成是类似的，都包括数据、运算、控制和传输这四种表示成分。其中：

数据：描述程序所涉及的数据对象。

在程序运行过程中，其值不变的数据称为常量，其值可以改变的数据称为变量；另外，有些可以不加任何说明就能引用的运算过程，称为标准函数，其函数值可以像常量或变量一样参加运算；由常量、变量、函数、运算符和圆括号组成的式子称为表达式，它在程序中代表一个值。程序设计语言所提供的数据结构是以数据类型的形式表现的，程序中的每一个数据都属于某一种数据类型。

运算：描述程序中应该执行的数据操作。

在程序中的运算一般都包括算术运算、关系运算和逻辑运算。

控制：描述程序的操作流程控制结构。

在程序中只要有三种形式的流程控制结构（即顺序、选择、重复结构），就足以表示出各种各样复杂的算法过程，这已从理论上得到证明。

传输：表达程序中数据的输入和输出。

每一种程序设计语言都含有赋值、输入和输出语句。

附录 1.2 程序设计的步骤

使用计算机解决具体问题时，通常需要下列几个步骤：

分析问题：首先是明确要解决什么问题，确定所需的输入、处理和输出对象，把解题过程归纳为一系列的数学步骤，建立各种量之间的关系。

算法设计：对所确定的解题模型提出解决的方法和步骤，选择适当的计算方法加以实现。算法设计过程中还要考虑数据的组织形式，即数据结构。

画出流程图：流程图是描述算法的常用工具，可以作为编写程序的依据来使用。

编写程序：在确定算法和画出流程图后，用选定的程序设计语言编程。

程序调试：用测试数据对编好的程序上机调试，分析所得的运行结果，特别要注意程序中的逻辑错误。反复调试，直到运行结果正确。

文档整理：在程序设计的各个步骤中，都要注意建立文档资料，内容包括任务要求、算法、流程图、程序清单、输入/输出数据的内容及格式、出错处理方法等。在程序设计完成时，文档资料应全部建立完毕。文档也是软件的一个组成部分。

附录 1.3 ASCII 码表

L \ H	0000	0001	0010	0011	0100	0101	0110	0111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	A	q
0010	STX	DC2	“	2	B	R	B	r
0011	ETX	DC3	#	3	C	S	C	s
0100	EOT	DC4	\$	4	D	T	D	t
0101	ENQ	NAK	%	5	E	U	E	u
0110	ACK	SYN	&	6	F	V	F	v
0111	BEL	ETB	,	7	G	W	G	w
1000	BS	CAN	)	8	H	X	H	x
1001	HT	EM	(	9	I	Y	I	y
1010	LF	SUB	*	:	J	Z	J	z
1011	VT	ESC	+	;	K	[	K	{
1100	FF	FS	,	<	L	\	L	
1101	CR	GS	-	=	M	]	M	}
1110	SO	RS	.	>	N	^	N	~
1111	SI	US	/	?	O	_	O	DEL

习题与实践

- 1. 什么叫程序？什么叫程序设计？
- 2. 何谓算法？简述程序设计的步骤。
- 3. 程序设计语言一般包括哪四种成分？
- 4. 试述面向对象程序设计的特点。
- 5. 写出 VC++ 的开发过程。
- 6. 上机调试运行课本的 zhu1_1.cpp 程序，熟练编译运行 C++ 程序的操作。
- 7. 运行下面的程序，记录发现的错误信息，并把它改正后再运行：

```
include <iostream.h>
int main()
{
    cout << "Hello World!"<<endl;
```

```
return 0;
```

8. 阅读下面的程序，不运行该程序而分析它的结果，然后上机验证你的分析是否正确。

```
#include<iostream.h>
```

```
int main()
```

```
{
```

```
    int x, y;
```

```
    x=5;
```

```
    y=7;
```

```
    cout<<"x+y="<<x+y<<endl;
```

```
    return 0;
```

```
}
```

9. 以下叙述哪一个是不正确的？

A) 一个 C++ 程序可由一个或多个函数组成。

B) 一个 C++ 程序必须包含一个 main() 函数。

C) C++ 程序的基本组成单位是函数。

D) 在 C++ 程序中，注释说明只能位于一条语句的后面。