



新编 21 世纪高职高专计算机系列规划教材

C++

程序设计应用教程

北京希望电子出版社 总策划

黄锦祝 主 编

黄锦祝 莫治雄 梁 裕 编 著



 科学出版社
www.sciencep.com



新编 21 世纪高职高专计算机系列规划教材

C++

程序设计应用教程

北京希望电子出版社 总策划

黄锦祝 主 编

黄锦祝 莫治雄 梁 裕 编 著

11312
11312



科学出版社
www.sciencep.com

内 容 简 介

本书根据两年制计算机高职教育的特点编写。

本书是面向对象及程序设计的基础教程，全书共分两篇。第1篇主要介绍程序设计的基本概念、C++的数据类型、常用运算符及各种表达式；C++的各种控制语句，包括条件选择语句及循环控制语句；C++扩展数据类型，包括数组、结构体、共用体、枚举类型及自定义类型；C++的指针及函数的定义与应用。第2篇主要介绍面向对象程序设计的基本概念、面向对象的基本特征，类及对象的定义与应用，函数重载、继承与派生、多态类及输入输出流等内容。

本书每章都列举了大量的实例，同时在章末安排了上机实训内容，真正做到上机指导与理论相结合。本书可作为两年制或三年制高职教育计算机及相关专业的程序设计课的教材，也可作为面向对象程序设计自学人员的参考书。

需要本书或需要得到技术支持的读者，请与北京中关村 083 信箱（邮编：100080）发行部联系，电话：010-82702660, 010-82702658, 010-62978181 转 103 或 238，传真：010-82702698，E-mail：tbd@bhp.com.cn

图书在版编目 (CIP) 数据

C++程序设计应用教程/黄锦祝主编。—北京：科学出版社，2005.5

(新编 21 世纪高职高专计算机系列规划教材)

ISBN 7-03-015128-3

I . C... II. 黄... III. C 语言—程序设计—高等学校：技术学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字 (2005) 第 017074 号

责任编辑：刘海芳 / 责任校对：叶子

责任印刷：媛明 / 封面设计：梁运丽

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码：100717

<http://www.sciencep.com>

北京市媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005 年 5 月第 一 版 开本：787×1092 1/16

2005 年 5 月第一次印刷 印张：14

印数：1~3 000 字数：332 000

定价：22.00 元

新编 21 世纪高职高专计算机系列规划教材

编 委 会 成 员 名 单

主任 高 林

副主任 谢玉声 袁启昌

胡伏湘 陆卫民

委 员 (排名不分先后)

莫治雄	曾 涛	刘习华
陈 晴	常荆燕	王 彦
万世明	丁文华	杜华兵
张 波	贺占魁	李建利
王路群	汪建立	徐建华
梁 裕	杨殿生	郑明红
黄锦祝	林锦章	杨 波
明志新	刘继清	胡新和
郑勇杰	曹 毅	殷均平

前 言

随着科学技术的飞速发展，以计算机技术和数字通信技术为核心的信息产业得到了快速发展，需要大量适应这一产业发展的技术应用性人才，也就是需要一大批掌握计算机知识并且熟悉计算机操作技能的高级技术人才。而从事计算机及相关行业的技术人才，一般都有必要学会并熟练掌握一门计算机语言。C++是当前最流行的面向对象程序设计语言之一，而且与C语言完全兼容，因此，很多高校计算机及相近专业都开设C++课程，作为软件开发、算法设计、操作系统等后续课程的基础。

近年来，我国高等职业教育发展很快。高等职业教育是我国实现高等教育大众化的重要手段，也是为生产、管理、服务第一线培养技术应用性人才的重要途径。随着信息产业的快速发展，国家对高等职业教育的计算机应用及软件技术专业非常重视。2003年专门成立了35所软件学院，2004年又确定79所高职院校作为计算机应用与软件技术技能型紧缺人才培养培训基地。另外，逐步将高等职业教育的学制由三年改为两年，这不仅缩短了培养周期，也促使高职院校必须加大课程体系改革，加大课程整合力度，体现有别于普通高等教育的高职特色。

两年制高等职业教育培养周期短，但是培养的学生既要符合高等教育的要求，也要满足职业教育的要求。因此，学生既要有“够用”的理论基础，又要具有适应工作岗位的操作技能，这就要求教材内容既要有一定的理论深度，又要具有强化学生技能的实训操作。本书是作者在从事多年高等职业教育C和C++教学过程中总结高等职业教育的特点并结合两年制高等职业教育程序设计基础必须具备的操作技能编写而成的。

本书分为两篇。第1篇主要介绍程序设计的基础知识、C++的数据类型、程序设计的基本结构、程序设计的基本方法等；第2篇主要介绍面向对象程序设计的基本概念、面向对象的基本特征、类及对象的定义与应用。本书理论与实训紧密结合，重点培养学生的实际动手能力，每章都列举了大量例子，同时在章节后面安排了一定数量的实训内容。

本书内容组织合理，层次分明，由浅入深，符合学习的一般规律，既可以作为两年制高等职业教育程序设计的教材，也可以作为面向对象程序设计的自学人员的参考书。

参加本书编写的有黄锦祝、莫治雄、梁裕、周会国、周晓芳，全书由黄锦祝拟定提纲和统一定稿，莫治雄对本书的编写进行具体的指导。

如果需要本书课后习题解答，请发邮件至yyy668@126.com。

编 者

类库 (OCL) 出入门群 简介
C++ 基本语义
类与对象
继承与派生
重载
指针
流

类与对象
类与对象
类与对象
类与对象
类与对象
类与对象
类与对象

第1篇 基础篇

第1章 C++程序设计语言概述	1
1.1 基础知识	1
1.2 应用举例	5
1.3 归纳与思考	6
1.4 上机训练	7
1.5 习题	8
第2章 C++的基本数据类型	10
2.1 基础知识	10
2.2 应用举例	23
2.3 归纳与思考	25
2.4 上机训练	25
2.5 习题	26
第3章 C++运算符与表达式	28
3.1 基础知识	28
3.2 应用举例	38
3.3 归纳与思考	39
3.4 上机训练	39
3.5 习题	41
第4章 C++控制语句	43
4.1 基础知识	43
4.2 应用举例	56
4.3 归纳与思考	58
4.4 上机训练	58
4.5 习题	59
第5章 C++扩展数据类型	60
5.1 基础知识	60
5.2 应用举例	74
5.3 归纳与思考	77
5.4 上机训练	77
5.5 习题	78
第6章 指针	80

目 录

6.1 基础知识	80
6.2 应用举例	88
6.3 归纳与思考	91
6.4 上机训练	91
6.5 习题	92
第7章 函数	93
7.1 基础知识	93
7.2 应用举例	106
7.3 归纳与思考	108
7.4 上机训练	109
7.5 习题	110
第2篇 提高篇	
第8章 面向对象程序设计概论	112
8.1 基础知识	112
8.2 归纳与思考	114
第9章 类和对象	115
9.1 基础知识	115
9.2 应用举例	134
9.3 归纳与思考	137
9.4 上机训练	137
9.5 习题	140
第10章 重载	146
10.1 基础知识	146
10.2 应用举例	152
10.3 归纳与思考	154
10.4 上机训练	154
10.5 习题	156
第11章 继承与派生	160
11.1 基础知识	160
11.2 应用举例	167
11.3 归纳与思考	168
11.4 上机训练	168

11.5	习题	171
第12章 多态类		177
12.1	基础知识	177
12.2	应用举例	182
12.3	归纳与思考	183
12.4	上机训练	184
12.5	习题	187
13	类成员函数	188
14	类成员函数	188
15	基类指针	189
16	派生类指针	190
17	虚函数	190
18	纯虚函数	191
19	多态图	191
20	归纳与思考	192
21	上机训练	193
22	习题	193
23	类成员函数	194
24	基类指针	194
25	派生类指针	195
26	纯虚函数	195
27	多态图	196
28	归纳与思考	196
29	上机训练	197
30	习题	197
31	类成员函数	198
32	基类指针	198
33	派生类指针	199
34	纯虚函数	199
35	多态图	200
36	归纳与思考	200
37	上机训练	201
38	习题	201
39	类成员函数	202
40	基类指针	202
41	派生类指针	203
42	纯虚函数	203
43	多态图	204
44	归纳与思考	204
45	上机训练	205
46	习题	205
47	类成员函数	206
48	基类指针	206
49	派生类指针	207
50	纯虚函数	207
51	多态图	208
52	归纳与思考	208
53	上机训练	209
54	习题	209
55	类成员函数	210
56	基类指针	210
57	派生类指针	211
58	纯虚函数	211
59	多态图	212
60	归纳与思考	212
61	上机训练	213
62	习题	213
63	类成员函数	214
64	基类指针	214
65	派生类指针	215
66	纯虚函数	215
67	多态图	216
68	归纳与思考	216
69	上机训练	217
70	习题	217

第13章 输入/输出(I/O)流类		192
13.1	基础知识	192
13.2	应用举例	212
13.3	归纳与思考	213
13.4	上机训练	214
13.5	习题	216
71	类成员函数	217
72	基类指针	217
73	派生类指针	218
74	纯虚函数	218
75	多态图	219
76	归纳与思考	219
77	上机训练	220
78	习题	220
79	类成员函数	221
80	基类指针	221
81	派生类指针	222
82	纯虚函数	222
83	多态图	223
84	归纳与思考	223
85	上机训练	224
86	习题	224
87	类成员函数	225
88	基类指针	225
89	派生类指针	226
90	纯虚函数	226
91	多态图	227
92	归纳与思考	227
93	上机训练	228
94	习题	228
95	类成员函数	229
96	基类指针	229
97	派生类指针	230
98	纯虚函数	230
99	多态图	231
100	归纳与思考	231
101	上机训练	232
102	习题	232
103	类成员函数	233
104	基类指针	233
105	派生类指针	234
106	纯虚函数	234
107	多态图	235
108	归纳与思考	235
109	上机训练	236
110	习题	236
111	类成员函数	237
112	基类指针	237
113	派生类指针	238
114	纯虚函数	238
115	多态图	239
116	归纳与思考	239
117	上机训练	240
118	习题	240
119	类成员函数	241
120	基类指针	241
121	派生类指针	242
122	纯虚函数	242
123	多态图	243
124	归纳与思考	243
125	上机训练	244
126	习题	244
127	类成员函数	245
128	基类指针	245
129	派生类指针	246
130	纯虚函数	246
131	多态图	247
132	归纳与思考	247
133	上机训练	248
134	习题	248
135	类成员函数	249
136	基类指针	249
137	派生类指针	250
138	纯虚函数	250
139	多态图	251
140	归纳与思考	251
141	上机训练	252
142	习题	252
143	类成员函数	253
144	基类指针	253
145	派生类指针	254
146	纯虚函数	254
147	多态图	255
148	归纳与思考	255
149	上机训练	256
150	习题	256
151	类成员函数	257
152	基类指针	257
153	派生类指针	258
154	纯虚函数	258
155	多态图	259
156	归纳与思考	259
157	上机训练	260
158	习题	260
159	类成员函数	261
160	基类指针	261
161	派生类指针	262
162	纯虚函数	262
163	多态图	263
164	归纳与思考	263
165	上机训练	264
166	习题	264
167	类成员函数	265
168	基类指针	265
169	派生类指针	266
170	纯虚函数	266
171	多态图	267
172	归纳与思考	267
173	上机训练	268
174	习题	268
175	类成员函数	269
176	基类指针	269
177	派生类指针	270
178	纯虚函数	270
179	多态图	271
180	归纳与思考	271
181	上机训练	272
182	习题	272
183	类成员函数	273
184	基类指针	273
185	派生类指针	274
186	纯虚函数	274
187	多态图	275
188	归纳与思考	275
189	上机训练	276
190	习题	276

第1篇 基础篇

第1章 C++程序设计语言概述

随着信息技术的飞速发展，计算机的应用遍及社会的各个领域。每个计算机工作者除了掌握计算机的基本操作外，都有必要熟练掌握一门编程语言。C++语言是在C语言的基础上发展起来的面向对象程序设计语言，是当今世界上最流行的面向对象程序设计语言之一。熟练掌握C++面向对象程序设计语言，不仅可以使用C++开发各种应用软件，还可以比较容易地理解、掌握其他面向对象程序设计语言。

要点：

- ◆ C++语言的发展史
- ◆ C++的主要特点
- ◆ C与C++的异同
- ◆ C++的程序结构
- ◆ C++运行环境

重点内容：

C++程序的结构及基本程序的书写方法，C++程序的调试方法。

1.1 基础知识

1.1.1 C++语言的发展历史

1972年，美国贝尔实验室的Dennis Ritchie在B语言的基础上开发出了C语言。最初的C语言只是为了描述和实现UNIX操作系统。它不但功能强大、可移植性好、使用方便，而且具有汇编语言的高效、快速、灵活等特点，又支持结构化程序设计。随着UNIX的广泛使用，C语言逐渐被人们认可和使用，到了20世纪80年代已经广泛流行，成为一种应用最广泛的程序设计语言之一。

虽然C语言功能强大，很受程序设计者的欢迎，但是也存在一些局限，主要是C语言不支持代码重用、不适合开发大型软件。随着计算机应用领域的不断拓展，计算机软件的规模越来越大、复杂性越来越高，C语言的局限性很快地表现出来。

1980年，美国贝尔实验室的Bjarne Stroustrup博士及其同事在C语言的基础上，引入面向对象的特征，开发出了C++语言。C++语言继承了C语言的精髓，扩充了对开发大型软件非常有效的面向对象功能，解决了C语言的不支持代码重用问题，为熟悉C语言的程序设计者提供功能更为强大的程序设计语言。

1.1.2 C++语言的特点

- (1) C++语言保持与C兼容，很多C代码不经修改就可以在C++环境中使用。
- (2) 用C++编写的程序可读性更好，代码的结构更合理。
- (3) C++支持面向对象的机制，具有现代面向对象语言的特点，同时又具有面向过程语言的功能，有利于软件开发人员从面向过程的编程方法到面向对象编程方法的转变。
- (4) C++软件具有可重用性及可扩充性，有很好的继承功能。
- (5) C++生成的代码质量比较高，运行效率仅比汇编语言代码慢10%~20%。

1.1.3 C与C++程序的比较

C++语言是在C语言的基础上发展起来的，所以其基本结构与C语言相同，所使用的数据类型、标识符基本一致，对于以前学习与使用C语言的程序开发人员来说，非常容易学习与理解，下面给出两个整数相加的C及C++源程序。

例1.1

```
/* add.c */
#include<stdio.h> /* 包括输入、输出预处理的头文件 */
void main() /* 主函数 */
{
    int x,y,sum; /* 定义三个整型变量 */
    x=10,y=25; /* 给x、y两个变量赋值 */
    sum=x+y; /* 把x加上y的值赋给sum */
    printf("sum=%d\n",sum); /* 使用函数printf输出sum变量的值 */
}
```

程序运行结果为：

sum=35

例1.2

```
// add.cpp
#include<iostream.h> // 包含输入、输出预处理的头文件
void main() // 主函数
{
    int x,y,sum; // 定义三个整型变量
    x=10,y=25; // 给x、y两个变量赋值
    sum=x+y; // 把x加上y的值赋给sum
    cout<<"sum="<<sum<<endl; // 使用cout输出sum变量的值
}
```

程序运行结果为：

sum=35

例1.1与例1.2的功能是一样的，都是定义三个变量，都是给x、y两个变量赋值，然后把两个变量的值相加，赋给sum变量，最后一行都是输出sum的值。

从上面两个例子，读者可以看出，C与C++的结构非常相似，都具有一个main()函数，编写的程序都放在一对大括号{.....}构成的范围中，整型变量的定义都是使用int标识符，每个语句都是用分号(;)结束。

C++程序与C程序也有区别。一般C的文件扩展名用“.c”表示，而C++的文件扩展名用“.cpp”表示；C程序的注释用“/*.....*/”表示，而C++可以用“/*.....*/”或者“//”表示；C的输出是使用printf函数完成，而C++使用cout输出流完成。

由于C++与C完全兼容，例1.1可以在C++环境中运行。

1.1.4 C++程序的结构特点

为了使读者对C++程序结构有更深入的了解，下面再看一个例子。

例1.3 求两个整数的最大值。

```
// max.cpp
#include<iostream.h>
int mymax(int x,int y) // 自定义函数mymax()
{
    int z;
    if (x>=y) z=x; // 判断如果x大于等于y，则把x赋给z
    else z=y; // 否则把y赋给z
    return z; // 返回z的值到调用函数
}
void main() // 主函数
{
    int x,y,max;
    cout<<"Please enter two integers:"; // 提示输入两个整数
    cin>>x>>y; // 从键盘输入两个整数
    max=mymax(x,y); // 调用 mymax() 函数，把返回值赋给max
    cout<<"The max is:"<<max<<endl; // 输出最大值
}
```

程序运行结果为：

Please enter two integers: 12 30

The max is: 30

通过例1.3可以看出C++程序的结构有以下特点：

(1) C++程序由函数构成，函数是C++程序的基本单位。例1.3由main()和mymax()两个函数组成，其中名为main()的函数称为主函数。一个能够独立运行的C++程序必须有一个而且仅有一个main()函数，main()函数可以放在程序的任何地方，但是无论放在哪里，程序运行时第一个执行的函数必定是主函数。此外，C++程序还可以包括系统提供的库函数，也可以是用户自己定义的函数，例1.3中的mymax()就是自定义的函数。

(2) C++函数由函数的说明部分和函数体两部分组成。

函数的说明部分包括函数类型、函数名、函数参数。函数类型规定为函数返回值的类

型，如 `int`、`char`、`float` 等。如果无返回值的函数，可以使用 `void` 类型。

函数体就是在函数说明部分下面的大括号 {……} 内的部分。

(3) C++ 中每个语句和数据定义必须以分号 (；) 结束

(4) C++ 程序书写格式非常灵活，一行内可以写几个语句，一个语句可以分在多行上。

(5) 在 `cout` 输出流中用 `endl` 换行。

(6) 可以使用 `/*……*/` 或 `//` 对 C++ 程序中的任何部分注释。由 `/*……*/` 括起来的程序段或放在 `//` 后面同一行的内容，程序将不执行，一般对程序段注释使用 `/*……*/`，对某一行进行注释使用 `//`。

1.1.5 C++ 的运行环境

世界上有很多种 C++ 语言，比较流行的有 Visual C++ 系列（简称 VC++）和 Borland C++ 系列，它们有一致的语法规则，但也有不同的操作规范及库函数为了能够满足不同用户的要求，本书所举的例子均可以在 VC++ 环境和 Borland C++ 环境中运行。下面通过一个简单实例介绍如何在 VC++ 环境及 Borland C++ 的早期版本 Turbo C++ 环境中实现。

例 1.4 编制一个程序，输入一个圆的半径，求出圆面积的程序。

设圆的半径为 r ，则圆的面积 s 为： $s = \pi r^2$

下面是在 VC++ 环境中实现的方法。

(1) 运行 Visual C++。

(2) 在 File 菜单下选择 New 命令，打开 New 对话框。

(3) 选择 Projects 选择项，并指定下列选项：

- Projects Win32 Console Application
- Project Name Mycircle
- Location 指定应用程序存放的位置（可以先建立一个专用的目录）
- Create New Workspace 选中（默认）
- Platforms Win32 选中（默认）

设置完成后，单击 OK 按钮，打开 AppWizard 对话框。

(4) 在 AppWizard 对话框中，选择 A Simple Application。

(5) 单击 Finish 按钮，显示 New Project Information 对话框，单击右下方的 OK 按钮。

至此，自动创建了一个主文件 Mycircle.cpp，项目还自动创建了预编译头文件 Stdafax.h 和系统设置程序 Stdafax.cpp。

(6) 选择右边的 Fileview 工作区，在 Source Files 中选择 Mycircle.cpp，可以见到 Mycircle.cpp 的内容如下：

```
// Mycircle.cpp: Defines the entry Point for the console application.
#include "stdafx.h"

int main(int argc, char *argv[])
{
    return 0;
}
```

(7) 在 Mycircle.cpp 中插入编写好的程序。最后完整的 Mycircle.cpp 程序如下：

```
// Mycircle.cpp: Defines the entry Point for the console application.
#include "stdafx.h"
#include<iostream.h>
int main(int argc, char *argv[])
{
    const float pi=3.14159;
    int r;
    float s;
    cout<<"Please enter a number:";
    cin>>r;
    s=pi*r*r;
    cout<<"s="<

```

到现在为止，整个程序已经编辑完成，运行程序即可。

下面是在Turbo C++环境中实现的方法。

- (1) 在安装好的Turbo C++中，选择BIN目录下的TC.EXE。
- (2) 在File菜单下选择New命令，自动生成一个Noname.cpp空文件。
- (3) 在编辑区中直接输入下面的程序段：

```
#include<iostream.h>
void main()
{
    const float pi=3.14159;
    int r;
    float s;
    cout<<"Please enter a number:";
    cin>>r;
    s=pi*r*r;
    cout<<"s="<

```

(4) 运行程序即可。

1.2 应用举例

例1.5 编一程序，实现求两个实数之和。

```
#include<iostream.h>
void main()
{
    float x,y,sum;
```

```

x=18.3;
y=25.9;
sum=x+y;
cout<<"sum="<

```

程序运行结果：

```
sum=44.2
```

分析：

(1) 程序中要求实现两个实数之和，所以变量x、y、sum必须定义为实型，否则将得不到正确的结果。

(2) C++程序的书写非常灵活，可以采用不同的书写形式，如本例中的几个语句

```
x=18.3;
```

```
y=25.9;
```

```
sum=x+y;
```

可以改成如下形式：

```
x=18.3,y=25.9,sum=x+y;
```

例1.6 编一程序，求两个实数之商

```
#include<iostream.h>
void main()
{
    float x=16,y=3;
    float m;
    m=x/y;
    cout<<"m="<

```

程序运行结果：

```
m=5.33333
```

分析：

(1) C++程序允许在定义变量的同时给变量赋值，称这种赋值方式为变量的初始化。

(2) 在使用变量时，要注意变量的类型。

1.3 归纳与思考

C++是在传统的C语言的基础上发展起来的，与C语言具有很高的兼容性，熟悉C语言的用户很容易上手。C++继承了C语言灵活的书写格式和较快的运行速度。

C++程序由函数构成，不管C++有多少个函数，总是从main()函数开始执行。C++的每个函数必须用大括号({})括起来，每一个语句必须以分号(;)结束。

C++语言与C语言及其他程序设计语言有什么不同？程序的结构有什么区别？

在您使用的C++环境中，如何编辑、运行C++的程序？如何对程序进行调试和修改？

1.4 上机训练

训练一 在C++开发环境中编辑、调试、运行程序

1. 训练目的

(1) 了解C++开发环境(开发环境可以根据读者的情况自由选择使用VC++或Turbo C++)。通过调试程序逐步掌握使用C++环境编辑源程序的方法。

(2) 掌握如何在C++环境中编译、运行程序。

2. 训练内容

(1) 编制、调试例1.4。

(2) 编制、调试程序, 输入正方形的边, 求正方形的周长及面积。

3. 训练步骤(假设在VC++环境中)

1) 进入VC++的过程

(1) 在Windows的中运行Visual C++。

(2) 进入Visual C++开发环境。

(3) 在File菜单下选择New命令, 打开New对话框。

(4) 选择Projects选择项, 并指定选项:

Projects Win32 Console Application

(5) 在界面的右边Project Name栏内填入Mycircle, 其他项使用默认值。

设置完成后, 单击OK按钮, 打开AppWizard对话框。

(6) 在AppWizard对话框中, 选择A Simple Application。

(7) 单击Finish按钮, 显示New Project Information对话框, 单击右下方的OK按钮。

至此, 自动创建了一个主文件Mycircle.cpp, 项目还自动创建了预编译头文件Stdafx.h和系统设置程序Stdafx.cpp。

(8) 选择右边的Fileview工作区, 在Source Files中选择Mycircle.cpp。

(9) 把Mycircle中的内容改变如下:

```
#include "stdafx.h"
#include<iostream.h>
void main()
{
    const float pi=3.14159;
    int r;
    float s;
    cout<<"Please enter a number:";
    cin>>r;
    s=pi*r*r;
    cout<<"s="<<s<<endl;
}
```

2) 运行程序

(1) 在菜单Build中找到 !Execute Mycircle.exe或在工具栏中找到“!”符号，用鼠标单击它，或者直接按键盘的Ctrl+F5。

(2) 当屏幕出现运行界面时，在Please enter a number:后面输入5。

```
Please enter a number: 5
```

(3) 按回车键后出现

```
s=78.5397
```

```
Press any key to continue
```

(4) s=78.5397就是运行的结果。

3) 修改程序

(1) 将上面程序的float s; 改为

```
int s;
```

重新运行程序，观察程序的运行结果。

(2) 输入下列程序，并运行之，观察程序运行结果。

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

```
void main()
```

```
{
```

```
int x=28, y=7;
```

```
int t;
```

```
m=x/y+x;
```

```
cout<<"m="<<m<<endl;
```

} 将x改为40，再运行程序，观察程序运行结果。

思考：如何将程序进行修改，使之能够得到正确的结果？

4) 重新建立新程序

(1) 将上面运行的VC++全部关闭，重新运行VC++，在新建的文件中，在Project Name中输入Mylx，其他按上面步骤进行操作。

(2) 编写程序，输入正方型的边，求出正方型的周长及面积。

1.5 习题

(1) C++语言的特点有哪些？

(2) 一个完整的C++程序包括哪几部分？

(3) 分析下列程序的运行结果：

```
#include<iostream.h>
void main()
{
```

```

int x,y,z;
float average;
x=12,y=32,z=19;
average=(x+y+z)/3.0;
cout<<"average="<

```

(4) 练习使用cout输出下列的内容:

Hello! How are you!

I am a student.

(5) 编一程序, 从键盘输入三个数给x,y,z, 输出它们的最小值。

第2章 C++的基本数据类型

在程序设计中，需要用到大量的数据。数据是程序处理的对象，由于应用领域的不同，需要各种不同类型的数据。C++预定义了非常丰富的数据类型，也允许用户自定义数据类型，满足不同程序设计的需要。本章先介绍常用的基本类型，比较复杂的类型及自定义类型将在后面的章节中介绍。

要点：

- ◆ C++语言基本符号组成
- ◆ C++基本的数据类型
- ◆ 各种数据类型的常量及变量
- ◆ C++数据类型的转换

重点内容：

C++各种类型数据的区别和使用方法。

2.1 基 础 知 识

2.1.1 C++符号表示

1. C++字符集

编写程序就像写文章一样，需要用到各种文字及符号。而这些文字及符号都是由最基本的符号组成，这些基本符号构成一个字符集。C++语言也有一个字符集，主要包括以下一些字符：

- (1) 大小写英文字母 A~Z, a~z
 - (2) 数字字符 0~9
 - (3) 特殊字符 + - * / \ = ' " () [] { } ! # % & ~ , . ^ < >
- | 空格

2. 符号

为了区分世界上的物体，我们都给这些物体进行命名，如桌子、汽车、木头等。在程序中，需要用到各种各样的变量名、符号常量名、函数名、数组名、文件名等，为了区分它们，需要给每一个命名不同的名字，这些名字就叫符号。C++的符号必须满足有关规则：

(1) C++的符号只能由字母、数字和下划线三种字符组成，而且第一个字符必须为字母或下划线，如sum, average, _name, x1, y3等都是合法的符号。

不合法的符号如12m, M.x, #xy, i*j等。

(2) C++的符号区分大小写字母，如 A 和 a Class 和 class 都是不同的两个名称

(3) 符号不能是系统预定义的关键字，如 int, if, for, return 等。