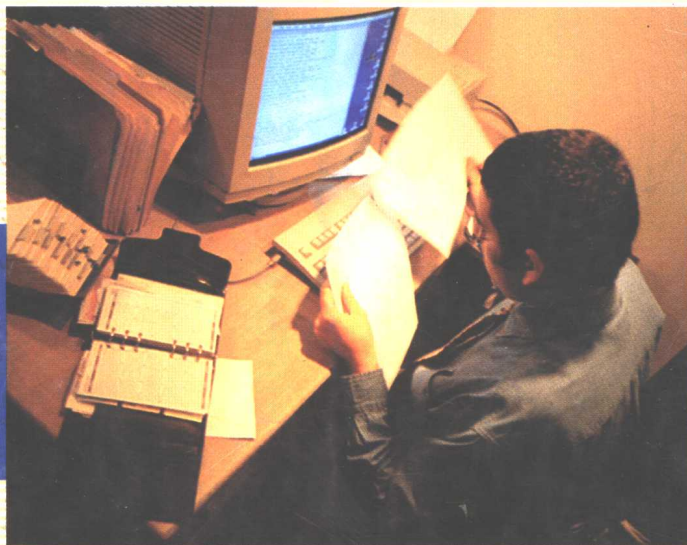


C/C++ 程序设计 实验指导与习题

刘振鹏 马胜甫 主编



河北大学出版社

责任编辑:马 力
封面设计:赵 谦
责任印制:蔡进建

图书在版编目(CIP)数据

C/C++ 程序设计实验指导与习题/刘振鹏等主编.
保定:河北大学出版社,2003.1
ISBN 7-81028-912-8

I .C... II .刘... III .C 语言—程序设计—自学参
考资料 IV .TP312

中国版本图书馆 CIP 数据核字(2003)第 002605 号

出版:河北大学出版社(保定市合作路 88 号)

印制:河北〇五印刷厂

印张:13

版次:2002 年 12 月第 1 版

字数:316 千字

经销:全国新华书店

规格:1/16(787mm×1092mm)

印数:0001~8000 册

印次:2002 年 12 月第 1 次

ISBN 7-81028-912-8/TP·46

定价:18.00 元

内容简介

本书是学习 C/C++ 语言程序设计的辅助基础教程,精心设计了 11 个实验单元,并对《C/C++ 程序设计》教材的各章习题做了详尽的解答,通过实验内容和剖析习题总结程序设计的要点。书中所给出的每个程序都有详细的注释,便于读者阅读和理解。通过实验内容和习题解答,所涉及的知识点不仅包括 C/C++ 语言的基础知识、基本规则及编程方法,以及 C++ 的面向对象的重要特征,如类、封装、继承、多态、I/O 流操作等,还包括 Windows 编程基础知识、应用 Windows API 函数的可视化编程技术要点以及应用 MFC 进行可视化编程的方法等。

本书特别强调加强基础、注重实践,采用循序渐进、逐步深入的方法,突出重点,注意将难点分开,通过实例化例程让读者全面、灵活、系统地掌握面向对象程序设计的技术思想和开发方法。

本书可作为大专院校各专业程序设计的实验教材和复习参考书,又可作为研究生计算机基础教育的辅助教材,也可供计算机培训班和读者自学使用。

前 言

C/C++ 程序设计语言是一门实践性很强的课程,若要真正掌握程序设计技术,使编程能力有实质性的提高,必须在认真听课并读懂教材的基础上,通过上机实验加强程序技术的基本技能训练。只有勤学苦练才能积累宝贵的编程经验,悟出程序设计技术的要领,牢固地掌握像 Visual C++ 这样的优秀的应用程序开发工具。

本书的作者长期从事计算机专业核心课程和程序设计语言课程的教学工作,并利用 C、C++、Visual C++ 等语言开发了多个软件项目,有着丰富的教学经验和较强的科研能力,对 C/C++ 有着较深入的理解。

本书根据《C/C++ 程序设计》教材的内容而编写。为了使读者能够掌握 C/C++ 程序设计语言的使用方法,并初步具备使用 C/C++ 程序设计语言开发应用程序和解决实际问题的能力,作者本着加强基础、注重实践、勇于创新、突出应用的原则,精选了 C/C++ 的内容,设计了 11 个实验单元,对教材中的习题大部分做了详尽的分析,给出较为完整的答案。力求使本书达到可读性、先进性、适用性和实用性的良好统一。

由于目前的计算机操作平台多为 Windows 图形界面平台,同时也为了与教材配套,本书以 Visual C++ 6.0 为程序调试与运行环境。本书中所给出的 C++ 程序示例均在 Visual C++ 6.0 环境下通过调试和运行。

本书在编写过程中得到了各位专家的热心指导与无私帮助,编者在此表示衷心的感谢。此外在本书写作时,还参考了大量文献资料,在此也向这些文献资料的作者深表感谢。感谢河北大学出版社马力、韩勇两位编辑,他们的辛勤工作使得本书能够在短时间内高质量的完成。

由于时间仓促和水平所限,书中难免有不当和欠妥之处,敬请各位读者不吝批评指正。

编 者

2002 年 10 月

目 录

上篇 实验指导

实验一	Visual C++ 6.0 的基本用法	(3)
实验二	数据类型、运算符和表达式	(10)
实验三	逻辑结构程序设计	(15)
实验四	循环结构程序设计	(20)
实验五	数组	(24)
实验六	指针	(29)
实验七	函数	(32)
实验八	结构体和联合体	(38)
实验九	类与对象	(44)
实验十	继承与虚函数	(57)
实验十一	重载与文件 I/O	(70)
Visual C++	实验环境介绍	(76)

下篇 习题与解答

第 1 章	绪论	(85)
第 2 章	数据类型及表达式	(88)
第 3 章	结构化程序设计	(90)
第 4 章	指针与引用	(105)
第 5 章	函数与预处理	(114)
第 6 章	结构体、联合体、枚举类型	(141)
第 7 章	面向对象设计	(161)
第 8 章	类与对象	(163)
第 9 章	继承与派生	(166)
第 10 章	运算符重载	(168)
第 11 章	输入/输出流	(193)
参考文献		(200)

上 篇

实 验 指 导

实验一 Visual C++ 6.0 的基本用法

一、实验目的

- ①了解和使用 Visual C++ 集成开发环境。
- ②熟悉 Visual C++ 6.0 环境的基本命令和功能键,熟悉常用的功能菜单命令。
- ③学习使用 Visual C++ 6.0 环境的帮助。
- ④学习完整的 C++ 程序开发过程。
- ⑤理解简单的 C++ 程序结构。
- ⑥了解用 Visual C++ 开发 Windows MFC 应用程序的过程。

二、实验设备与环境

1. 硬件配置

586 以上 PC 兼容机或品牌机,配有彩色显示器、鼠标、键盘,内存不小于 32MB,硬盘自由空间不少于 60MB。推荐配置为内存 64MB(或以上),硬盘自由空间 500MB 以上。

2. 软件配置

操作系统:Windows 98、Windows 2000、Windows NT 4.0 以上版本。

开发集成环境:Microsoft Visual C++ 6.0 以上版本。

三、实验内容与步骤

本次实验将学习有关 Visual C++ 开发环境的一些知识,并尝试实现一个简单的 DOS 程序和 Windows 程序。通过本次实验,可以了解用 Visual C++ 开发 C++ 应用程序的过程。

1. 熟悉 Visual C++ 实验环境

- (1)启动 Developer Studio,看看初始化界面由哪些部分组成。
- (2)查看各菜单项,看看都有哪些子菜单和命令。
- (3)将鼠标放置于各工具条图标上,系统会自动显示该图标代表的命令含义,了解一下都有哪些命令。
- (4)在任意工具条上单击鼠标右键,弹出式菜单上将显示所有可用的工具条,选择其中没有对号(✓)的项,看看有什么效果,再选择有对号(✓)的项,又有什么效果?
- (5)将鼠标移动到任意工具条上,将鼠标放到图标间隙,按下鼠标左键不放,移动鼠标到屏幕中间,有什么现象发生?再将它拖回到原来位置,有什么现象发生?
- (6)将鼠标移动到左边的工作区窗口,按下鼠标左键不放,移动鼠标到屏幕中间,有什么现象发生?再将它拖回到原来位置,有什么现象发生?
- (7)将鼠标移动到下边的输出窗口,按鼠标右键,弹出一个菜单,选择其中的菜单项“Hide”,结果如何?要重新显示该窗口,选择菜单项“View”→“Output”,窗口是不是又显示

出来了?

(8)学习使用帮助系统。选择菜单项“Help”→“Contents”,启动 MSDN 联机帮助系统,学习使用该帮助系统。联机帮助系统是一个相对独立的程序,它和 Developer Studio 是两个程序,但是它的启动和停止都受 Developer Studio 影响。MSDN 联机帮助系统运行的前提条件是 Developer Studio 在运行。

(9)选择菜单项“File”→“Exit”退出 Developer Studio。

2. 控制台应用

用 AppWizard 建立一个控制台应用,在终端上输出“Hello”。

术语:“控制台应用程序”是一个在 DOS 窗口中运行的基于字符的程序。由于这种模式的应用程序比 Windows 程序简单,先选择利用 Visual C++ 来建立这样一个应用,这样使得读者可以将精力先投入到学习使用 C++ 编程语言,而不需要把过多的精力投入到学习复杂的 Windows 编程中去。

(1)创建第一个应用

首先创建一个项目(project),项目将代表用户的应用,存放用户应用的所有信息,包括源文件、资源文件、编译链接设置等。创建项目的步骤为:

- ① 启动 Developer Studio。
- ② 从主菜单中选择菜单项“File”→“New”,将显示出“New”对话框。
- ③ 选择“Projects”标签,并从列表中单击“Win32 Console Application”选项。
- ④ 在“Location”编辑框中输入工作目录名称,如“C:\student\your_name”(由实验指导教师指定)。
- ⑤ 在对话框的右上角的“project name”编辑框内键入项目的名字,如“Hello”,系统将自动为用户的项目分配一个默认的目录。
- ⑥ 单击“OK”按钮继续。
- ⑦ 如果是 Visual C++ 6.0,系统将显示一个询问项目类型的程序向导,选择“an empty project”选项。

⑧ 单击“Finish”按钮或“OK”按钮结束配置,创建应用程序。

这时系统为用户创建一个新的项目,并且在左边的工作区窗口中将出现用户项目的名字。工作区窗口除原来的“InfoView”标签外又增加了两个标签(如果是 Visual C++ 6.0,则没有“InfoView”标签):“ClassView”和“FileView”。“ClassView”从类的角度显示项目中建立的各个类,双击某个类名将会在右边的文档显示区显示类的定义文件并把文件的当前位置定位到所选的类;“FileView”显示构成项目的各个文件,选择某一文件将会在右边的文档显示区显示文件内容;“InfoView”是 Visual C++ 5.0 的帮助文件目录,在这里可以选择所要获取帮助的标题,在右边将显示帮助内容。

(2)编辑一个简单的 C++ 源程序

用下面的方法在创建的项目中添加一个文件:

- ①菜单上选择菜单项“File”→“New”。
- ②在“New”对话框中选择“File”标签,单击“C++ Source File”选项。

③选中“Add to Project”复选框。

④在右边的“File name”编辑框中为文件指定一个名字,如“Hello”,系统将自动加上后缀“cpp”。

新的空白文件将自动打开,显示在文档显示区。在文件中输入以下内容:

```
#include <iostream.h>
int main( )
{
    cout << "hello,world!" << endl;
    return 0;
}
```

仔细检查输入的内容,确保内容正确。

(3)保存源文件

单击工具栏中的“save”图标,或者选择菜单项“File”→“Save”来保存源代码文件。

C++ 源文件的扩展名为 cpp。扩展名非常重要,Developer Studio 根据文件的扩展名来区分文件类型,并且根据文件类型提供相应的编辑帮助(如正确的语法高亮度显示)。

(4)编译、链接得到可执行程序

编辑结束后,仔细检查输入的内容,看有无错误。确认没有错误之后,选择菜单项“Build”→“Build Hello.exe”来编译所创建的项目(也可以按功能键 F7)。如果输入的内容没有错误,那么,在屏幕下方的输出窗口将会显示:

```
hello.exe - 0 error(s), 0 warning(s)
```

如果在编译时得到错误或警告,是源文件出现错误,再次检查源文件,看是否有错误,改正它。

(5)改正源程序中的错误

编译的错误会在 Developer Studio 的下方的输出窗口显示出来,逐个查看这些错误的内容,用鼠标双击,光标可以自动移动到发生错误的源程序的相应位置,仔细检查输入的源程序,改正发生错误的地方,注意是否遗留了分号、引号或括号等。改正后,再重复步骤(4)的操作,直到编译链接通过为止。

(6)运行程序

可以有三种方式运行用户的程序:

①在开发环境中运行程序。

选择菜单项“Build”→“Execute hello.exe”(或者 Ctrl+F5),在开发环境中执行用户的程序。程序运行以后将显示一个类似于 DOS 的窗口,在窗口中输出一行“hello,world!”,紧接着在下面显示“Press any key to continue.”,这句话是系统提示用户按任意键退出当前运行的程序,回到开发环境中。按任意键,窗口关闭,退回到 Visual C++ 开发环境。

②在 DOS 环境下运行程序。

打开 DOS 窗口,改变工作路径到项目目录,该目录是用户在创建目录时指定的。如果不记得了,可以在 Developer Studio 中的工作区窗口中选择项目名称(这里是“hello files”),

然后选择菜单项“View”→“Properties”,将可以显示出项目路径。

切换到“debug”子目录下,运行“hello.exe”,程序将输出“hello,world!”。

③在 Windows 环境下运行程序。

打开 Windows 的资源管理器,找到程序所在的目录,运行它。程序将输出“hello,world!”。

3. Windows 应用

用 AppWizard 建立一个 MFC Windows 应用,在窗口中输出“Hello,World!”。

术语:AppWizard 是一个工具,利用该工具,可以创建一个建立在 MFC 基础上的窗口应用程序框架,然后在这个框架中加上自己的应用逻辑。可以选择所创建的应用类型,最常用的是多文档应用(就像字编辑器 Microsoft Word 一样,可以同时打开多个文档窗口的应用)、单文档应用(类似于 Windows 提供的 notepad,一次只能打开一个文档)和对话框应用(类似于 Windows 的时钟程序)。

(1) 创建一个新项目

利用 Developer Studio 的 AppWizard 创建一个新的项目,步骤为:

①选择菜单项“File”→“New”,系统将显示“New”对话框。

②选择“Projects”标签,在显示的项目类型中选择“MFC AppWizard(exe)”选项。

③在右边的“Project Name”编辑框中输入项目名称,如“helloMFC”,然后按“OK”按钮。

④MFC AppWizard 将分几步询问有关要建立的新项目的配置。第一个对话框问是创建哪种类型的应用(单文档、多文档还是对话框类型),选择创建单文档应用“Single document”,然后按“Next”按钮。

⑤翻过后面的五个页面(按“Next”按钮),每个页面可以改变项目的不同选项,这个例子暂时不设置这些选项。

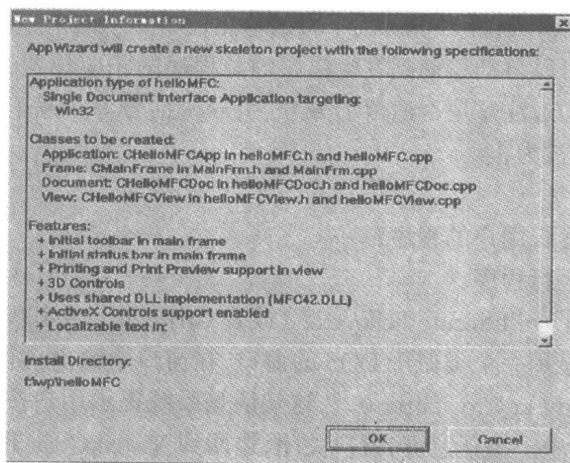


图 1-1 新建项目信息

⑥最后一个 MFC AppWizard 屏幕告诉用户 AppWizard 为用户自动产生的类。单击“Finish”按钮,AppWizard 显示一个关于该项目的摘要,列出这些类和用户所选择的特征,如图 1-1 所示。

⑦单击“OK”按钮,系统自动产生 helloMFC 所需要的文件。

(2) 浏览 helloMFC 项目

当用 MFC AppWizard 创建了 helloMFC 项目后,这个项目的工作区窗口将会打开,工作区窗口除了原来的“InfoView”以外,增加了三个标签:“ClassView”、“ResourceView”和“FileView”,如图 1-2 所示,其中“ClassView”显示工作区中所有项目的类及类的成员;“ResourceView”显示项目中包含的资源文件;“FileView”显示项目中的各种文件资源。

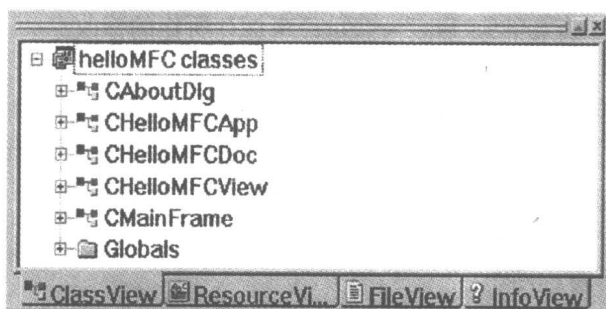


图 1-2 ClassView 示意图

可以先选择“FileView”看一下 AppWizard 为用户创建了哪些文件,然后选择“ClassView”看一下定义了哪些类。“ClassView”中还可以看到一个“Globals”文件夹,单击它前面的加号,可以看到,有一个预定义的全局变量“theApp”,这是用户的 Windows 应用程序类的对象。

(3) 编译链接运行

按功能键 F7 或者选择菜单项“Build”→“Build helloMFC.exe”,编译链接得到可执行程序,再按 Ctrl+F5 或者选择菜单项“Build”→“Execute helloMFC.exe”运行该程序。程序的结果如图 1-3 所示。

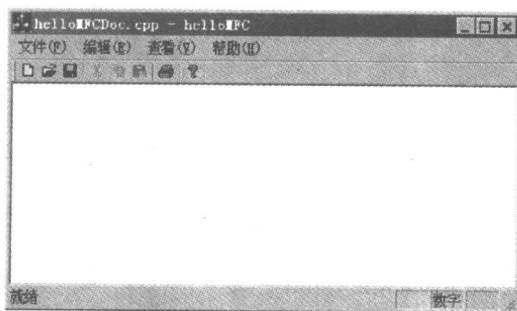


图 1-3 程序执行结果

(4)用 MFC 处理输出

如果希望在程序中间的窗口上显示一行文字“Hello, How do you do? ”。应该如何修改呢?

①在工作区窗口中选择“ClassView”标签,单击“helloMFC classes”前面的加号(如果已经变成减号则不做此操作)。

②单击“CHelloMFCView”类前面的加号。

③双击 OnDraw 函数,在右边的文档将显示窗口显示文件“helloMFCView”的内容,并且自动将光标定位到 OnDraw 函数处。

④修改 OnDraw 函数的定义,在最后一行加一句:

```
pDC->TextOut(50,50,"Hello,How do you do?");
```

⑤按 Ctrl + S 或者选菜单项“File”→“Save”来保存所作的修改。

(5)编译链接并运行

重新编译链接该项目,运行程序。可以用 Ctrl + F5 直接运行程序,系统将询问是否重新编译该项目,回答“是(Yes)”,如果有编译错误,仔细检查所加的一句话是否有错。当编译链接通过后,系统会自动运行该程序。结果与上面有什么不同?

经过上述修改后,程序可以输出一行文字“Hello, How do you do? ”。

4. 编写简单的计算程序

输入圆的半径,计算圆的周长和面积并输出。

(1)创建一个控制台项目

选择菜单项“File”→“New”,在“Projects”标签下选择“Windows32 Console Application”选项,输入项目名称“circle”,然后单击“OK”按钮。

(2)在项目中增加一个文件

选择菜单项“File”→“New”,在“Files”标签下选择“C/C++ Source File”选项,输入文件名称“circle”,然后单击“OK”按钮。

(3)在文件中输入以下内容

该程序让用户输入圆的半径 Radius,输出圆的周长 Perimeter 和面积 Area。

```
#include <iostream.h>
#include <math.h>
const double PI = 3.14159; // 定义一个常量 PI

int main( )
{
    int radius;
    double perimeter, area;
    cout << "Please input the radius: ";
    cin >> radius;
    perimeter = 2 * PI * radius;           // 周长 = 2πR
    area = PI * pow(radius, 2);           // 面积 = πR2
```

```

    cout << "The perimeter of the circle : " << perimeter << endl;
    cout << "The area of the circle : " << area << endl;
    return 0;
}

```

注:power(radius,2)表示求 radius 的平方,power(x, y)是系统预定义的函数,该函数计算 x 的 y 次方。该函数的原型在文件 math.h 中说明。

(4)编译、链接并运行程序

如果输入的程序有误,查找并修改错误,直到编译链接通过。运行程序,测试数据。

[测试数据]

输入:2

输出:

The perimeter of the circle : 12.5664

The area of the circle : 12.5664

输入:10

输出:

The perimeter of the circle : 62.8318

The area of the circle : 314.159

[思考问题]

- ①程序中为什么要将头文件 math.h 包含进来?
- ②建立控制台应用程序的通用步骤是怎样的?

[常用编辑命令]

虽然许多编辑命令可以通过菜单和工具栏实现,但大量的编辑命令都可以通过键盘实现。以下命令通常用键盘实现:

- ①撤消前一次操作。当进行了一次错误的操作时,可以通过敲击键盘上的 Ctrl + Z 完成。
- ②重复前一次操作。通过敲击 Ctrl + Y 实现。
- ③剪切一行。用 Ctrl + L 来删除一行并将它放到剪贴板中。
- ④剪切。将选中的文本删除并将它放到剪贴板中,用 Ctrl + X 实现。
- ⑤复制。将选中的文本复制到剪贴板中,用 Ctrl + C 实现。
- ⑥粘贴。将剪贴板中的内容放到编辑器中文本的当前位置处(由光标指示),用 Ctrl + V 实现。

要想了解关于键盘操作命令的完整列表,可以选择“Help”菜单下的“Keyboard Map”选项。没有必要记住所有的命令,有些根本不常用。

实验二 数据类型、运算符和表达式

一、实验目的

①掌握 Visual C++ 基本数据类型的字节宽度和范围表示,熟悉如何定义一个整形、字符型和实型的变量,以及对它们赋值的方法。

②掌握不同的类型数据之间赋值的规律。

③学会使用 C++ 的有关算术运算符,以及包含这些运算符的表达式,特别是自加(++)和自减(--)运算符的使用。

④进一步熟悉 C++ 程序的编辑、编译、链接和运行的过程。

二、实验设备与环境

1. 硬件配置

586 以上 PC 兼容机或品牌机,配有彩色显示器、鼠标、键盘,内存不小于 32MB,硬盘自由空间不少于 60MB。推荐配置为内存 64MB(或以上),硬盘自由空间 500MB 以上。

2. 软件配置

操作系统:Windows 98, Windows 2000, Windows NT 4.0 以上版本。

开发集成环境:Microsoft Visual C++ 6.0 以上版本。

三、实验内容与步骤

1. 基本数据类型的长度

编写一个程序,输出基本数据类型 char、short、int、long、float、double 和指针类型 void *、char *、short *、int *、long *、float *、double * 的数据类型的长度。

[实现要求]

搞清所使用系统上运行的 C++ 编译器中每个基本数据类型的长度。

[实现提示]

利用函数 sizeof(数据类型名)来得到各个数据类型的长度。

[思考问题]

为什么所有的指针长度一样?

2. 整形、字符型变量的定义及赋值

(1)输入并调试运行下面的程序

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

void main( )
{
    char c1, c2;
    c1 = 'a';
    c2 = 'b';
    cout << c1 << setw(5) << c2 << endl;
}
```

①运行此程序。

②在此基础上增加一个语句：

```
cout << (int)c1 << setw(5) << (int)c2 << endl;
```

再运行,并分析结果。

③将第 1 行改为：

```
int c1, c2;
```

再使之运行,并观察结果。

④再将第 2、3 行改为：

```
c1 = a;           // 不用单撇号
```

```
c2 = b;
```

再使之运行,分析其运行结果。

⑤再将第 2、3 行改为：

```
c1 = "a";         // 用双撇号
```

```
c2 = "b";
```

再使之运行,分析其运行结果。

⑥再将第 2、3 行改为：

```
c1 = 300;         // 用大于 255 的整数
```

```
c2 = 400;
```

再使之运行,分析其运行结果。

(2)输入并调试运行下面的程序

```
#include <iostream.h>

void main( )
{
    char c1 = 'a', c2 = 'b', c3 = 'c', c4 = '\101', c5 = '\116';
    cout << c1 << c2 << "\t" << c3 << "\t"
        << c1 << c2 << c3 << endl;
```

```

    cout << "\t\b" << c4 << c5 << endl;
}

```

在实验前先分析程序,写出应得结果,实验时将二者对照。

(3)输入并调试运行下面的程序

```

#include <iostream.h>
#include <iomanip.h>

void main( )
{
    short a,b;
    unsigned c,d;
    long e,f;
    a = 100;
    b = -100;
    e = 50000;
    f = 32767;
    c = a;
    d = b;
    cout << a << setw(12) << b << endl;
    cout << c << setw(12) << d << endl;
    cout << e << setw(12) << f << endl;
    c = a = e;
    d = b = f;
    cout << a << setw(12) << b << endl;
    cout << c << setw(12) << d << endl;
}

```

请对照程序和运行结果分析:

①将一个负整数赋给一个无符号的变量,会得到什么结果。画出它们在内存中的表示形式。

②将一个大于 32767 的长整数赋给整形变量,会得到什么结果。画出它们在内存中的表示形式。

③将一个长整数赋给无符号的变量,会得到什么结果(分别考虑该长整数的值大于或等于 65535 以及小于 65535 的情况)。画出它们在内存中的表示形式。

同学们可以改变程序中各变量的值,以便进行比较。例如: a = 65580, b = - 40000, e = 65535, f = 65580。