

北京市高等教育精品教材立项项目配套教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

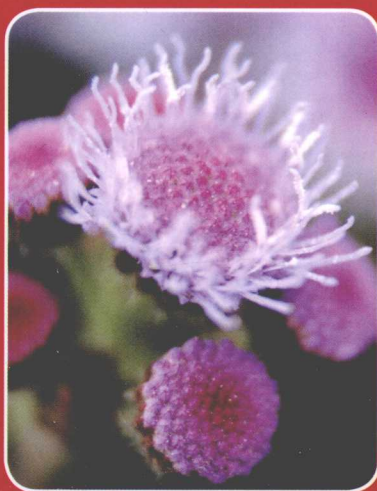
C++高级语言程序设计 案例与实践辅导

A Complete C++ Practice Guidance

徐惠民 主编

刘瑞芳 许桂平 肖波 赵衍运 杨文川 徐雅静 顾淑清 编著

- 一本书覆盖 C++ 各个实践教学环节辅导内容
- 丰富的编程案例提高 C++ 面向对象编程能力
- 完整的课程设计案例提高面向对象设计能力



精品系列



人民邮电出版社
POSTS & TELECOM PRESS

北京市高等教育精品教材立项项目配套教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C++高级语言程序设计 案例与实践辅导

A Complete C++ Practice Guidance

徐惠民 主编

刘瑞芳 许桂平 肖波 赵衍运 杨文川 徐雅静 顾淑清 编著



精品系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C++高级语言程序设计案例与实践辅导 / 徐惠民主编

— 北京 : 人民邮电出版社, 2012. 3

21世纪高等学校计算机规划教材. 精品系列

ISBN 978-7-115-27276-8

I. ①C… II. ①徐… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第264164号

内 容 提 要

本书是《C++高级语言程序设计》的学习辅导书。全书共 12 章, 第 1 章介绍了在 VC2005 集成开发环境下编程的步骤和程序调试方法, 第 2 章~第 9 章与主教材《C++高级语言程序设计》对应, 包括了原教材各章的习题答案、补充的程序设计案例和上机实验题, 第 10 章、第 11 章涉及课程设计的相关内容(第 10 章介绍了 Windows 程序设计的基础知识, 第 11 章提供了 3 个课程设计案例和设计题), 第 12 章是全国计算机等级考试 C/C++ 考试真题和详解。

本书的重点是编程案例和课程设计案例。本书编写的目的是为学生打好程序设计的基础, 提高其编程能力, 因此, 在编程案例的选择上特别注意涵盖 C++ 语言的基本概念, 不仅给出各案例的程序代码, 还附有分析、设计、代码说明、运行结果解释等。另外, 示例代码体现了对培养良好编程习惯的重视。

本书为读者学习 C++ 高级语言程序设计、提高编程能力提供了充足的内容, 适合作为大学各专业的 C++ 程序设计课程的辅导书和 C++ 课程设计的教材, 也可供程序设计爱好者自学使用。

北京市高等教育精品教材立项项目配套教材

21 世纪高等学校计算机规划教材

C++高级语言程序设计案例与实践辅导

◆ 主 编 徐惠民

编 著 刘瑞芳 许桂平 肖 波 赵衍运 杨文川

徐雅静 顾淑清

责任编辑 贾 楠

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

大厂聚鑫印刷有限责任公司印刷

◆ 开本: 787×1092 1/16

印张: 17

2012 年 3 月第 1 版

字数: 450 千字

2012 年 3 月河北第 1 次印刷

ISBN 978-7-115-27276-8

定价: 34.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

[illegible]

本书是《C++高级语言程序设计》的配套辅助教材。本书编写的主要目的,是希望提高学生使用 C++语言的程序设计能力,因此本书也是一本 C++编程指导和参考书。

全书分为 12 章,第 1 章介绍 VC2005 编程环境。现在的教学中,很多地方还在使用 VC6.0 编程环境。但是,VC2005 对于 C++标准的支持更全面。在介绍编程环境的同时,这部分更强调了调试程序的方法。

“习题答案”部分对于概念题和简答题的答案都写得很简单，主要是给出正确的思路。习题中一些比较典型的编程题，放到了“编程案例”部分进行分析和解答。其余的编程题也给出了解答。

“实验题”分为两种：基础题和编程题。基础题侧重概念，引导学生用实验的方法来解决概念问题；编程题的答案没有在书中给出，读者可以在人民邮电出版社教学服务与资源网（www.ptpedu.com.cn）上获得。老师可以根据教学的需要来使用题目。

本书最后一章提供了两份全国计算机等级考试 C++ 考试的真题和解析。由于

该试题主要是测试面向对象程序设计，之后还增加了一份全国计算机等级考试 C 语言考试的真题和解析，以补充对于面向过程程序设计的测试。这部分可以作为学生是否全面掌握 C++ 的自测卷。

本书的作者都是多年从事 C++ 教学和使用 C++ 语言进行项目开发的人员，可以说本书凝结了多人的智慧和心血。第 1 章、第 7 章、第 10 章、第 12 章由刘瑞芳编写，第 2 章由许桂平编写，第 3 章由肖波编写，第 4 章、第 5 章由赵衍运编写，第 6 章由杨文川编写，第 8 章、第 9 章由徐惠民编写，第 11 章由徐雅静编写，第 2 ~ 9 章的实验题由顾淑清编写。全书由徐惠民主编和统搞。

本书可以作为学习 C++ 程序设计的参考书，更可供编程爱好者自学使用。书中如有不当之处，欢迎广大读者批评指正。

编 者
2011 年 10 月

目 录

第 1 章 VC2005 编程环境介绍.....1	第 8 章 继承与派生.....127
1.1 VC2005 集成开发环境简介.....1	8.1 《C++高级语言程序设计》习题 答案.....127
1.2 建立 WIN32 控制台应用程序.....2	8.2 编程案例及参考例程.....131
1.3 程序调试方法.....5	8.3 实验题.....153
第 2 章 基本数据类型和表达式.....8	第 9 章 多态.....157
2.1 《C++高级语言程序设计》习题答案.....8	9.1 《C++高级语言程序设计》习题 答案.....157
2.2 编程案例及参考例程.....9	9.2 编程案例及参考例程.....159
2.3 实验题.....16	9.3 实验题.....178
第 3 章 C++控制语句.....19	第 10 章 Windows 程序设计 基础.....182
3.1 《C++高级语言程序设计》习题答案.....19	10.1 基于 Windows API 编程.....182
3.2 编程案例及参考例程.....23	10.2 基于 MFC 编程.....187
3.3 实验题.....32	第 11 章 课程设计.....195
第 4 章 数组及其他自定义类型.....36	11.1 彩色控制台应用程序.....195
4.1 《C++高级语言程序设计》习题答案.....36	11.1.1 案例分析：俄罗斯方块游戏.....195
4.2 编程案例及参考例程.....40	11.1.2 题目一.....206
4.3 实验题.....54	11.1.3 题目二.....207
第 5 章 函数.....56	11.2 MFC 应用.....207
5.1 《C++高级语言程序设计》习题答案.....56	11.2.1 案例分析：计算器设计.....207
5.2 编程案例及参考例程.....59	11.2.2 题目一.....218
5.3 实验题.....79	11.2.3 题目二.....218
第 6 章 指针和引用.....82	11.3 网络应用程序.....219
6.1 《C++高级语言程序设计》习题答案.....82	11.3.1 案例分析：邮件发送客户端.....219
6.2 编程案例及参考例程.....85	11.3.2 题目一.....230
6.3 实验题.....91	11.3.3 题目二.....231
第 7 章 类和对象.....96	第 12 章 全国计算机等级考试二级测试 及分析.....232
7.1 《C++高级语言程序设计》习题答案.....96	12.1 2011 年上半年 C++笔试题题及 答案.....232
7.2 编程案例及参考例程.....98	
7.3 实验题.....123	

12.2	2010 年下半年 C++ 笔试题题及	解答.....	252
	答案.....		242
12.3	2011 年上半年 C 语言笔试题题及		
		12.4 C++ 上机考试模拟题及解答.....	263

第 1 章

VC2005 编程环境介绍

本章介绍 VC2005 编程环境及其基本的使用方法。

1.1 VC2005 集成开发环境简介

较早期程序设计的各个阶段都要用不同的软件来进行处理，如先用字处理软件编辑源程序，再用编译程序进行编译，然后用链接程序进行函数、模块连接，开发者必须在几种软件间来回切换操作。为了使开发者便利、高效地开发各种应用程序，现在的编程开发软件集编辑、编译、调试等功能于一身，这样就大大方便了用户。目前大多数开发语言都有相应的集成开发环境（Integration Development Enviroment, IDE），IDE 是一种辅助程序开发人员开发软件的应用软件，一般包括代码编辑器、编译器、调试器和图形用户界面工具，集成了代码编写功能、分析功能、编译功能、调试功能等，这样的软件可以独立运行，也可以和其他程序并用。有些 IDE 还支持多种编程语言，但一般而言，IDE 主要还是针对特定的编程语言而量身打造。

Visual Studio 是一套完整的开发工具集，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C#和 Visual J#全都使用相同的集成开发环境（IDE），利用此 IDE 可以共享工具且有助于创建混合语言解决方案。这里仅介绍 Visual C++（VC2005）的使用，用于开发 C++程序。

VC2005 集成开发环境由如下若干元素组成：菜单栏、工具栏、停靠或自动隐藏在左侧、右侧、底部以及编辑器空间的各种工具窗口。在任何给定时间，可用的工具窗口、菜单和工具栏取决于所处理的项目或文件类型。启动界面如图 1-1 所示，窗口大致被划分为 3 部分，左侧是项目工作区窗口，右侧是编辑窗口，下侧是输出窗口。

可以从起始页开始工作，其中，最近的项目这一栏表示最近几次使用 VC2005 打开的 C++项目目录，鼠标单击其中之一即进入到该开发项目中，或者单击“创建 项目”建立一个新项目。也可以从菜单栏里的“文件→新建”或“文件→打开”开始工作。

MSDN 是一个 HTML 格式的帮助用户，可以提供在程序开发中所需的各种信息，包含的内容非常丰富，可以从菜单栏选择“帮助”进行查看和查找所需要的帮助，它还能脱离集成开发环境独立运行。

左侧的项目工作区窗口一般由解决方案资源管理器、类视图、资源视图 3 个标签组成，如图 1-2 所示。一个解决方案中可以包含几个项目，一个项目是一个完整的程序，可以编译、链接，生成可执行程序，所以在任一时刻，解决方案中的几个项目只能有一个是激活的。单击类视图标签可以看到各项目中的所有类，以树形结构显示，进一步单击相应的类进行编辑。单击资源视图标签

可以看到各项目中的所有资源列表。

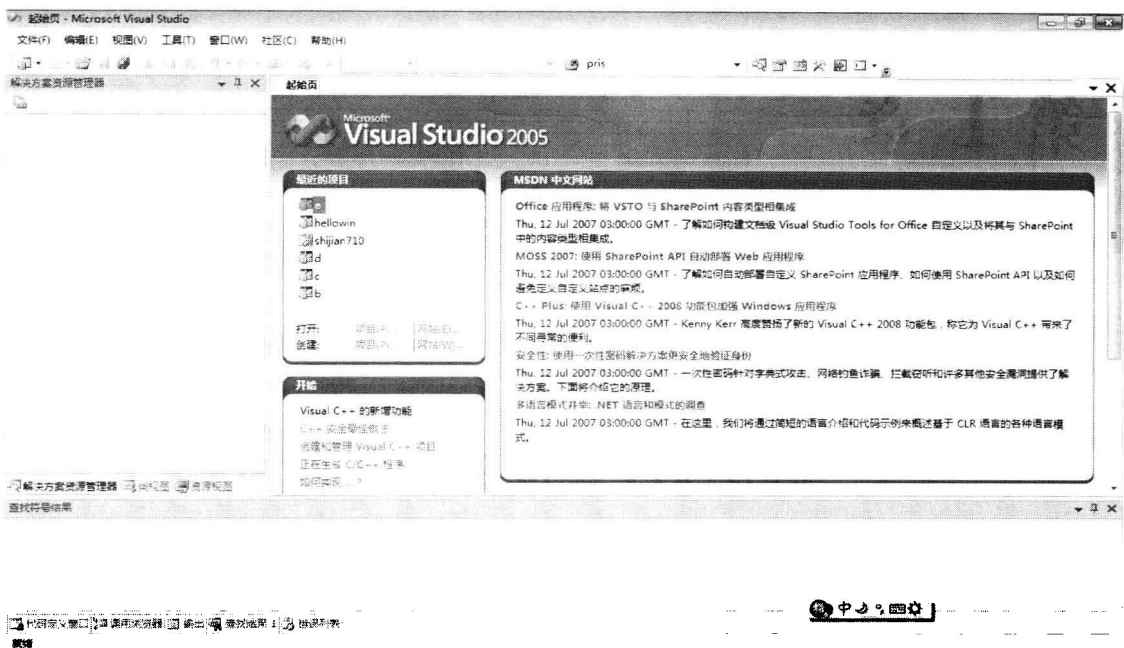


图 1-1 VC2005 开始界面



图 1-2 项目工作区

编辑窗口会显示当前可编辑的程序源文件或资源文件，可同时打开多个编辑窗口，它们以平铺或重叠方式显示，可以通过标签点选。

下侧的输出窗口，当编译、链接时会显示相关信息。

1.2 建立 WIN32 控制台应用程序

按照下面步骤建立一个 WIN32 控制台应用程序。

1. 在窗口中选择“文件→新建→项目”，会弹出如图 1-3 所示的窗口。

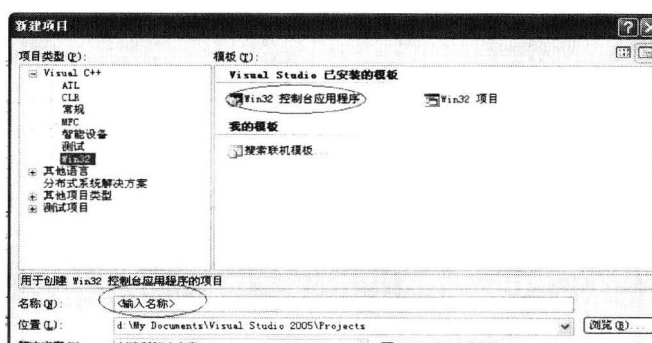


图 1-3 新建项目第一步

2. “输入名称”是输入工程的名称，不必加上后缀。按“确定”按钮后会出现如图 1-4 所示窗口，选择“下一步”按钮。

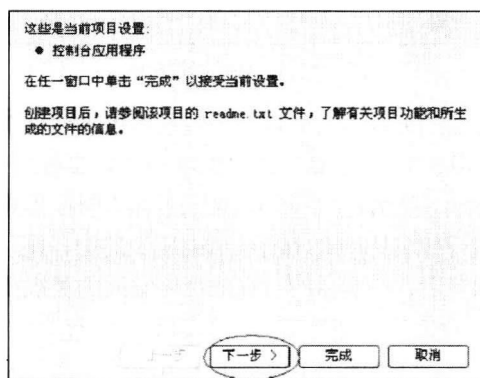


图 1-4 新建项目第二步

3. 在弹出的如图 1-5 所示的窗口中，选择“空项目”，最后按“完成”按钮。

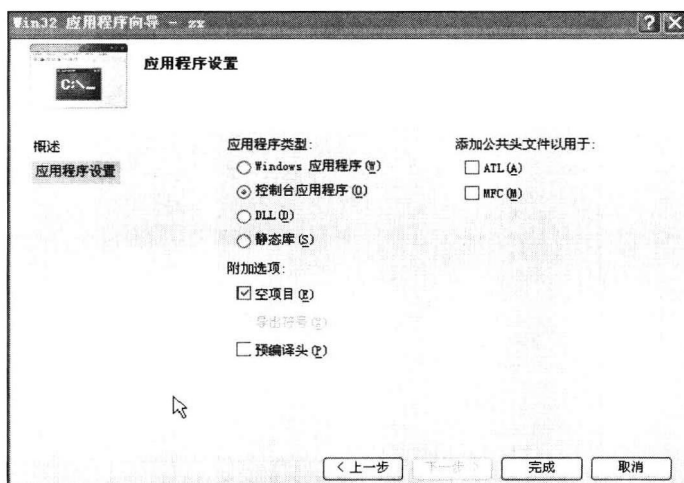


图 1-5 新建项目第三步

4. 这时候在“解决方案”上看到源文件,如图 1-6 所示。

5. 鼠标右键单击“源文件”后选择“添加”→“新建项”。然后弹出一个对话框,如图 1-7 所示。

6. 输入文件名,这个文件名就要加上后缀了。想写 C 程序就加上 C 后缀,写 C++ 程序后缀加不加都行,默认是.cpp 后缀。这时就可以在编辑窗口编写源文件了。



图 1-6 新建项目第四步

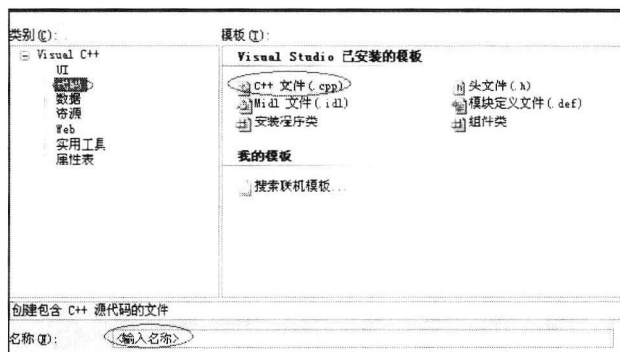


图 1-7 新建项目第五步

在编辑含有类的开发项目时,可以单击类视图标签查看各个项目下的类结构,即其数据属性与方法属性,如图 1-8 所示。其中,“宏与变量”即当前项目中定义的所有宏与变量;“全局 Typedef”即用户使用 Typedef 命令声明的一些别名;“全局函数与变量”即在当前项目下所有文件中都可见的函数和变量;一些三色方框对应的名称即在当前项目中定义的类;花括号“{}”表示当前项目中定义的命名空间,打开其下拉内容后,会展示所有在该命名空间的类视图,关于命名空间具体含义可上网查阅。用鼠标单击其中任意内容,即可查看该类别下的所有对象。例如,图 1-8 中选中了类“SiteTrade”,在其下方的方框中,分别展示了该类中声明的所有成员函数与数据属性,对应如下。

```
~SiteTrade(void)
load_file_to_data(sodict_buil...
SiteTrade(u_int size_of_idx...
mintf_filt_dup_dict
```



图 1-8 类视图窗口

函数与数据由不同的图标表示。使用鼠标左键双击对应方法或属性，则跳到源代码中对应的行，非常便于阅读较大型的程序。

1.3 程序调试方法

编程时程序有逻辑错误，但程序员又很难定位错误所在，这时往往需要调试程序。设置断点是调试程序最常用的技巧。断点是调试器设置的一个代码位置。当程序运行到断点时，程序中断执行，回到调试器。调试时，只有设置了断点并使程序回到调试器，才能对程序进行在线跟踪。下面介绍简单调试的步骤。

1. 添加断点

在头文件或源文件的当前行（光标所在行）设置断点，从菜单栏选择“调试→切换断点”或“调试→新建断点”命令，也可以直接按下快捷键“F9”。设置断点成功后，该行的最左侧会有一个红色的圆点，如图 1-9 所示。

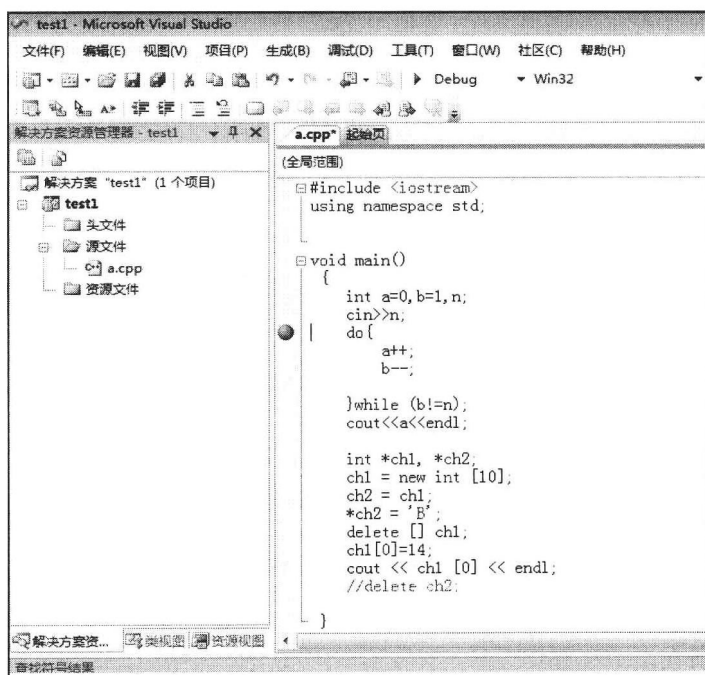


图 1-9 设置断点

2. 跟踪运行

从菜单栏选择“调试→启动调试”命令或按下快捷键“F5”，程序进入调试运行状态，执行到断点处会停下来，如图 1-10 所示。

可以继续跟踪，从菜单栏选择“调试→逐语句”命令或按下快捷键“F11”，进行单步跟踪，每次执行一条语句。也可以从菜单栏选择“调试→逐过程”命令或按下快捷键“F10”，则每次执行一条语句，遇到函数调用语句，直接得到函数运行结果，不会跟踪进入函数体。如果“逐语句”跟踪已经进入函数体，又觉得没必要跟踪函数的运行，可以从菜单栏选择“调试→跳出”命令或

按下快捷键“Shift + F11”，退出函数回到调用处。

3. 查看运行状况

跟踪运行的目的是想在程序执行过程中停下来，查看运行状况。当程序在断点处停下来或单步停下来时，可以单击下侧窗口的“局部变量”标签，查看所有变量的值，也可以查看堆栈、线程等状况。如果想把自己要关注的变量放在一个监视窗口里，可以在变量上单击鼠标右键，从弹出的快捷菜单里选择“添加监视”，如图 1-10 所示，以后单击“监视”标签查看自己所关注的变量。



图 1-10 查看运行状况

前面介绍的仅仅是简单的调试方法，任何时候都可以根据需要使用第一步、第二步或第三步，直到找到逻辑错误所在。

设置断点时也可以设置各种条件，以提高调试速度。对于新加的断点，可以单击鼠标右键，在弹出的快捷菜单中选择“断点→条件”命令，为断点设置一个表达式，当这个表达式发生改变时，程序就会被中断。也可以选择“断点→命中次数”命令设置让程序先执行多少次，再次到达断点时停下来，如图 1-11 所示。

下面列出常用的快捷键。

F5: 调试运行，遇到断点就停下来

Ctrl + F5: 开始执行（不调试）

Shift + F5: 停止调试

Ctrl + Shift + F5: 重启调试



图 1-11 条件断点

F9：切换断点

Ctrl + F9：启用/停止断点

Ctrl + Shift + F9：删除全部断点

F10：单步执行（跨过函数）

F11：单步逐句执行

Ctrl + F10：运行到光标处

使用快捷键，比起单纯使用鼠标进行点选操作，可以大大加快编程调试速度。

2.1 《C++高级语言程序设计》习题答案

2.1 答：有错的是

A12-3：变量名中不能有‘-’符号

123：变量名不能以数字开始

_ 123：变量名中不能有空格

2.2 答：只有表达式 C 不正确，不能将字符串赋值给一个字符变量。

2.3 答：32 位机中，int、float、double 类型在内存中分别占用 4、4、8 字节。

long double 变量在内存中占用 8 字节，和 double 相同。

2.4 答：字符串"ab\\n"在机器中占 5 个字节。

2.5 答：A

2.6 答：是 double 类型。

2.7 答：表达数学关系 $1 < a < 5$ 的 C++表达式可以是

$a > 1 \ \&\& \ a < 5$

$a == 2 \ || \ a == 3 \ || \ a == 4$

$!(a <= 1) \ \&\& \ !(a >= 5)$

2.8 答：常量是常数或代表固定不变值的名字，常量的内容初始化以后一直保持不变。变量的内容可以在运行过程中随时改变。变量要占用固定的内存地址，常量一般不占用内存，但是用 const 定义的常量也要占用内存。

2.9 答：关系运算符有 ==、>、<、>=、<=、!=

逻辑运算符有 !、&&、||

2.10 答：‘\089’是不正确的，因为 89 不是八进制数。

2.11 答：4

2.12 答：1.0

2.13 答：4.0

2.14 答：301

2.15 答：10

2.16 (1) 答：a1=1 a2=1

(2) 答:

0

2

10

8

10

(3) 答: 0

(4) 答:

****1

***12

**123

1****

12***

123**

2.17 答: 程序 1 的结果是 16 960。因为计算结果超过 short 的最大值, 溢出了。程序 2 的结果是 1 000。

2.18 参见补充题 2.1。

2.19 答: 0

因为 float 数本身的精度所限, 即使规定输出精度是 8 位, 也只能显示为 0。

定义为 double 类型后, 结果是 9.999 999 9e-008。

2.20 答: 运行结果为

FFFFFFBA

98FDEC45

98FDEC00

位运算实现的操作为

$x| \sim 0xFF$: 取 x 的最低字节, 其余位置为 1

$x \wedge 0xFF$: x 的最低字节按位取反, 其余位保持不变

$x \& \sim 0xFF$: x 的最低字节置为 0, 其余位保持不变

2.21 参见补充题 2.2。

2.2 编程案例及参考例程

例 2.1 以下程序通过 for 语句重复累加 0.01 共 10 次, 最后判定相加结果和 0.1 是否相等。请写出运行的结果, 并进行解释。

分析

(1) 程序中用一个 for 循环求 10 个 0.01 的累加和, 并将累加和存于 float 类型变量 sum 中, 即相当于完成: $sum = 0.01 + 0.01 + 0.01 + 0.01 + 0.01 + 0.01 + 0.01 + 0.01 + 0.01 + 0.01$ 。

(2) 程序中用 if 语句显示 sum 和 0.1 的比较结果, 如果表达式 $sum == 0.1$ 的结果为真, 则显示“相等”, 否则显示“不相等”。

程序

```
//例 2.1 浮点数的比较
#include <iostream>
using namespace std;
int main()
{
    float sum=0.0;
    for(int i=0;i<10;i++)
        sum=sum+0.01;
    cout<<sum<<endl;
    if(sum==0.1)
        cout<<"相等"<<endl;
    else
        cout<<"不相等"<<endl;
    return 0;
}
```

程序说明

(1) 运行结果:

```
0.1
不相等
```

(2) 解释: 通过分析, 表达式 $\text{sum}==0.1$ 虽然在数学上应该为真, 但在程序中却不为真。原因是浮点类型的有限精度导致在重复的累加操作中引入舍入误差, 所以在程序中的结果是不相等。



一般在检查两个浮点数是否相等时, 可以通过指定最大误差容限, 来检查这两个值是否足够接近, 而不是直接测试它们是否相等或不等。例如, 可以先指定最大误差容限: `const float Delta=0.000 01;` 然后把 $\text{sum}==0.1$ 改为: `fabs(sum-1.0)<=Delta;` 即可。其中使用了数学库函数 `fabs()`, 来返回浮点参数的绝对值。

例 2.2 把从键盘连续读取的多个字符串, 连接成一个字符串, 并输出到屏幕。每个字符串之间用空格隔开。

分析

(1) 因为程序中要处理字符串, 所以我们使用 `string` 类型的变量, 而且因为不知道输入多少个字符串, 所以当输入文件结束符时表示输入结束; 在 Windows 操作系统中, 以 “Ctrl+Z” 组合表示文件结束符。

(2) 因为程序中要输入多个字符串, 所以要使用循环语句 `while(cin>>str)`, 当输入字符串时, 提取运算符 “>>” 从 `cin` 提取字符串写入变量 `str` 中, 此时 `cin` 返回一个非 0 值, `while` 的循环条件为真, 执行循环体; 当输入文件结束符时, `while` 循环条件为假, 中止循环。

(3) 在循环体中, 按照题目要求把输入的字符串进行连接。

程序

```
//例 2.2 字符串的连接
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s_str, str;
```