高等学校计算机课程规划教材

C++与数据结构基础简明教程

陆　明
赵国瑞　主编
汪大菊

天津大学出版社
TIANJIN UNIVERSITY PRESS

高等学校计算机课程规划教材

C++与数据结构基础简明教程

陆明　赵国瑞　汪大菊　主编

天津大学出版社
TIANJIN UNIVERSITY PRESS

内 容 提 要

本书介绍了C++语言的主要内容，包括数据类型、程序控制、指针、函数、类、继承、重载、多态性、输入/输出等，还介绍了线性表、栈、队列、数组、树、二叉树、图等基本数据结构以及相应的算法和简单应用，对部分算法还给出了C++语言描述。

本书内容由浅入深、重点突出、概念清晰、通俗易懂，并含有大量的程序设计例题和各种练习题，以供读者自学。

本书适合作为普通高等学校非计算机专业少学时(60～80学时)的程序设计教材，还可供各类计算机软件人员和软件开发人员、程序设计爱好者和工程技术人员参考。

图书在版编目(CIP)数据

C++与数据结构基础简明教程/陆明，赵国瑞，汪大菊主编. —天津：天津大学出版社，2008.2

ISBN 978-7-5618-2612-6

Ⅰ.C... Ⅱ.①陆...②赵...③汪... Ⅲ.①C语言-程序设计-教材②数据结构-教材 Ⅳ.TP312

TP311.12

中国版本图书馆CIP数据核字(2008)第000417号

出版发行	天津大学出版社
出 版 人	杨欢
地　　址	天津市卫津路92号天津大学内(邮编：300072)
电　　话	发行部：022-27403647　邮购部：022-27402742
网　　址	www.tjup.com
短信网址	发送“天大”至916088
印　　刷	天津泰宇印务有限公司
经　　销	全国各地新华书店
开　　本	185mm×260mm
印　　张	20.5
字　　数	512千
版　　次	2008年2月第1版
印　　次	2008年2月第1次
印　　数	1－3 000
定　　价	35.00元

凡购本书，如有缺页、倒页、脱页等质量问题，烦请向我社发行部门联系调换

版权所有　　侵权必究

前　言

为适应少学时教学需要，本书在《C++程序设计与数据结构基础教程》的基础上进行了精编，去掉了原书中第5章、第11章，合并了第6章和第7章，并在基本保证完整性的前提下，对原书的内容进行了取舍。

C++语言是当今最流行的一种高级程序设计语言，它具有极高的灵活性、强大的功能和非常高的效率，常用于专业应用程序的开发，应用十分广泛。本书介绍了C++语言的主要内容，包括数据类型、程序控制、指针、函数、类、继承、重载、多态性、输入/输出等。数据结构是程序设计的重要理论与技术基础，掌握数据结构知识对于有效地开发计算机程序非常必要。本书介绍了线性表、栈、队列、数组、树、二叉树、图等基本数据结构以及相应的算法和简单应用，对部分算法还给出了C++语言描述。

本书各部分都提供了丰富的例题，读者可以从这些例题中学习程序设计的技巧和有效使用数据结构求解问题的方法。各章后设有习题，通过各种类型习题的练习，不仅能加深读者对基本概念和定义的理解，而且还能通过上机提高编程能力和调试程序能力。

本书建议授课学时数(含上机)为60～80学时。

本书第1章～第4章由汪大菊编写，第5章～第8章由赵国瑞编写，第9章～第11章由陆明编写。

由于作者水平有限，书中不足之处在所难免，敬请读者批评指正。

编者

目 录

试读结束：需要全本请在线购买： www.ertongbook.com

第 1 章　C++ 程序设计基础

本章主要介绍 C++ 语言及程序设计的基本概念，包括 C++ 简单程序格式，C++ 基本数据类型、运算符及表达式和简单的输入输出。通过本章的学习，可以掌握 C++ 语言的基础知识并能编写顺序结构的简单 C++ 程序，为后续的学习打下基础。

1.1　C++ 语言概述

1.1.1　C++ 语言与程序设计

语言是人类交流思想的工具。在人和计算机打交道的时候，要让计算机按人们预先安排的步骤进行工作，就要解决人和计算机进行交流的问题。人和计算机进行交流的语言，称为计算机语言。

最早的计算机语言称为机器语言。机器语言是一条条二进制代码的指令，每一条二进制指令表示一个功能，例如取数、加运算等。计算机可以直接执行机器语言而不需要转化。由计算机专业人员用指令编写的机器语言程序难读、难写、难修改。为简化机器语言，人们用符号代替二进制代码，这种便于记忆的符号语言称为汇编语言。用汇编语言写出的程序称为汇编源程序。汇编源程序上机运行时必须通过一个“汇编程序”将汇编源程序翻译成二进制指令机器才能执行。汇编语言的出现，为程序设计人员提供了很大方便。但用汇编语言编写程序同样是一件繁琐的工作，它需要程序设计人员了解计算机硬件的细节，因而影响了计算机的推广、应用。高级语言的出现为广大非计算机专业人员应用计算机提供了极大的方便。目前常用的高级语言有 BASIC、FORTRAN、C、C++ 及 JAVA 等。用高级语言编写的程序称为源程序。计算机不能直接执行源程序，必须经过“编译程序”或“解释程序”将源程序翻译成机器指令，机器才能执行。不同的高级语言有不同的编译程序或解释程序。因为计算机语言是程序设计使用的语言，所以又称为程序设计语言。

程序设计就是将解决某个问题的过程用程序设计语言描述出来，计算机按这个描述逐步实现。不同高级语言的程序设计方法不同。因此，从程序设计方法的角度看，高级语言中的 FORTRAN、C 等都是面向过程的结构化程序设计语言，而 C++ 、JAVA 等在面向过程语言的基础上增加了面向对象的语言内容，常称为面向对象的程序设计语言。面向对象的程序设计方法被认为是最有希望、最有前途的方法。它是为适应计算机发展，特别是为操作系统等软件资源的发展而产生的。面向对象的程序设计方法是对面向过程的结构化程序设计方法的一次革命。这两者的根本区别是：在面向过程的结构化程序设计中，程序设计人员把重点放在解决某个问题的过程上；而在面向对象的程序设计中，设计人员把着眼点放在解决“什么”问题上，而不是问题的解决过程上，也就是只需关心一个对象能做什么，而不必关心对象的内部构成，从而使程序设计人员以更开阔的视野来观察问题、解决问题，使计算机的求解过程更接近人的思维过程，从而可以更充分发挥计算机系统的潜在能力。

C++语言是在C语言基础上为支持面向对象的程序设计研制的程序设计语言。C语言的前身是1967年英国剑桥大学Matin Richards推出的BCPL语言。1970年美国贝尔实验室的Ken Thompson以BCPL为基础,设计出简单而又很接近硬件的B语言。1973年贝尔实验室的D.M.Ritchie在B语言的基础上保留了它简练、更接近硬件等特点,设计出了C语言。特别是用C语言成功地改写了UNIX操作系统,从而使C语言迅速得到推广,先后被移植到大、中、小型机及微型机上。C语言是一种优秀的程序设计语言,它既有高级语言的优点,也有低级语言的特点,多年来受到普遍欢迎。C++语言是C语言的扩充,它保留了C语言高效灵活、易于理解、可移植性好等优点,又克服了C语言类型检查不严格以及程序规模较大时很难查找和排除错误等缺点,同时增加了面向对象的特征,并扩充了许多新功能,使得C++语言成为目前最流行的程序设计语言之一。

由于C++语言与C语言兼容,从而使许多C程序代码和用C语言编写的大量的库函数不经修改就可以在C++语言中使用。正是由于C++语言是C语言的扩充而不是一种纯正的面向对象的语言,所以C++语言既支持面向过程的结构化程序设计,又支持面向对象的程序设计。从1980年由美国贝尔实验室的Bjarne Stroutstrup提出C++语言以后,几经修改完善,并于1998年颁布了ISO/MSIC++语言标准。目前,C++语言仍在不断发展之中。

1.1.2 C++语言和面向对象的程序设计

C++语言与C语言的最大区别是前者支持面向对象的程序设计方法。面向对象的程序设计方法简称OOP(Object Oriented Programming),是现代程序设计的里程碑,也是结构化程序设计以及模块化、数据抽象等方法的发展。因此,作为程序设计者,既要学习结构化程序设计方法,又要学习面向对象的程序设计方法。

在面向对象的程序设计中首先要理解"对象"一词的含义。客观世界中的任何事物都可称为对象。这些事物小到一个螺丝钉、一本书,大到一个工厂、一个学校。对象是具有某种特性和某种功能的东西。对于使用者来说,选择一个对象主要不是考虑对象内部是怎样构成的,而是考虑对象能做什么,也就是说它的功能是什么。就如同用户购买电视机时,只要知道怎样操纵按钮就行了,至于内部构造不必关心。面向对象是一种崭新的方法,用这种方法观察世界,将客观世界看成由许多不同种类的对象组成,每种对象有特定的特征和操作,不同对象的相互作用构成了完整的客观世界。把具有共同特征(属性)和操作(方法)的对象抽象出来便形成了类。类是面向对象的程序设计中最重要的概念。显然,对象是类的具体化,是某个类的一个实例。面向对象的程序主要是由对象和类建立起来的,具有封装性、继承性和多态性。

在面向对象的程序概念中,把数据和与这些数据有关的操作结合在一起,形成一个有机的整体,称为"封装"。封装是一种信息隐藏技术,通过封装将数据和其有关的行为隐蔽起来,而将一部分行为作为对外的接口。从用户或应用人员的角度看,对外的接口就是为其提供的操作功能,即为一组服务,而服务的具体实现(即对象的内部)却被隐藏着。正如一台电视机给人们提供的是外部操作,而它内部的电路却被隐藏起来了一样。在C++程序设计中封装是由类实现的。

"继承"是面向对象的程序设计中最重要的特征。继承所表达的是一种类之间的相互关系。它使某个类除了具有特有的属性和方法外,还可以继承其他类的属性和方法。在实际工作中,一个特定问题的认识和处理,往往前人已经进行过较为深入的研究和探讨,他们的研究

成果后人可以利用，并且在此基础上有所发展和创造，这正是社会发展的过程。C++ 提供的类的继承机制允许程序设计人员在保持原有类的基础上，通过继承进行扩充成为新的类。

“多态”也是面向对象的程序设计中的一个重要特征。在 C++ 程序中，程序设计人员用向一个对象发送消息来完成一系列动作。而多态性是指不同的对象接收到相同的消息时，会产生不同的动作。C++ 语言支持两种多态性，即编译时的静态多态性和程序运行时的动态多态性。静态多态性通过函数重载实现，动态多态性通过虚函数机制实现。

1.2　C++ 程序开发过程

一个 C++ 系统通常由程序开发环境、语言和 C++ 标准库等部分组成。

在 C++ 程序开发环境中，一个 C++ 程序要经过编辑(edit)、预处理(preprocess)、编译(compile)、连接(link)、装入(load)和执行(execute)等六个阶段。以使用 Windows 下的集成程序开发环境(IDE) Microsoft Visual C++ 6.0 为例，除了编辑源程序要由用户自己完成外，其他阶段只需要执行一个命令就可以由 C++ 系统自动完成。Windows 下的典型 C++ 程序开发环境的各阶段及其关系如图 1.1 所示。

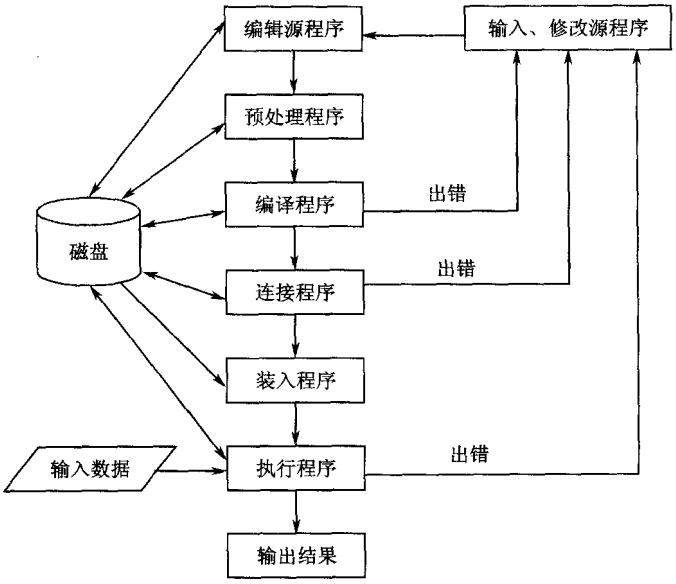

图 1.1　C++ 程序开发过程

在编辑阶段，使用编辑程序输入、修改源程序并存入磁盘中，文件名由用户指定。文件的扩展名一般为 .cpp(源程序)或 .h(头文件)。

预处理、编译两个阶段是用一个编译(compile)命令自动先后完成的。预处理阶段处理各种预处理命令，并生成一个临时 C++ 源程序文件存入磁盘中，交由编译阶段进行语法检查和语义分析。如果发现语法错误，则显示尽可能多的错误信息，以便用户查找和修改错误后再次编译。如果未发现语法错误，则将生成目标程序文件并存入磁盘(扩展名一般为 .obj)。

在连接阶段，连接程序把编译阶段生成的目标程序(可能有多个，一个 .cpp 文件生成一个目标文件)与 C++ 的标准库的代码拼装成一个完整的可执行程序文件并存入磁盘(扩展名一

般为.exe)。

装入程序把可执行程序从磁盘放入主存,并做好执行前的一切准备,如把代码段、各种数据段的首地址存入相关寄存器中。然后,在CPU的控制下执行该程序,遇到从键盘读入语句时,便等待用户输入数据并完成提交(一般按下回车键)。程序执行期间也可能存取磁盘中的外部数据文件。当程序执行中发生错误时,如除数为0及负数取对数或负数开平方时,计算机将显示出错信息。如果程序正常执行结束,则按程序中输出语句的要求输出结果。

在上述各阶段中都可能出现错误。其中语法错误是最常见的。这类错误可以由编译程序找出来,而且常常由于前面一个语法错误而引起后面很多语法错误。语法错误通过修改源程序很容易排除。程序运行期间也会出现致命错误而使程序中止执行(如负数取对数等)。有些错误通过仔细分析程序流程也比较容易查找和排除。最难以查找和排除的错误是属于程序逻辑上的错误(如不慎将"==”写为"=”或"<=”写为"<”)和程序运行时发生的非致命性错误(如数组下标越界)。因此在调试程序时,常常需要从后面的某个阶段返回到前面的编辑阶段查找错误。

1.3 C++程序实例

这里给出几个简单的C++程序实例,以便先对C++程序有感性认识。

例1.1 在屏幕上输出"Hello,you are welcome!"。

```
/* Hello   program */
 # include  < iostream.h >
void main()
{   cout << "Hello,you are welcome!";}                    //你好,欢迎你!
```

执行这个程序,将在屏幕上显示如下内容:

```
Hello,you are welcome!
```

程序说明如下。

①程序中第1行"/* Hello program */"为注释。在C++程序中有两种注释形式。一种是由"/*"开始到"*/"为止,中间的所有内容为注释。这种注释可以占若干行,可以出现在程序的任何地方。另一种注释形式是用"//"开始,"//"后的所有内容都是注释,该注释内容直到本行结束处终止。如例1.1程序第4行的"//你好,欢迎你!"。"//"可出现在程序的任何行后,也可单独占一行。一般前者常用于较长内容的注释,后者多用于较短内容的注释。注释是编程者以文字的方式对程序中有关代码进行的说明或解释,为阅读程序提供了方便,但编译系统并不处理它。

②程序第2行#include < iostream.h >是编译预处理行。其中"iostream.h"为一文件名。该文件是系统库文件,文件中定义了程序中使用的"cout"和"<<”操作运算的有关信息。"cout"和"<<”是系统提供的输出操作。当在程序中使用系统提供的输入、输出、数学函数运算等操作时,必须在程序开始处嵌入相关文件,称为包含文件或头文件。C++系统手册会给出各种库文件所提供的功能。预处理命令必须以"#”开头。因为预处理不是程序的语句,所以预处理行最后没有";”。这样的行必须独占一行。

③从第3行开始的结构

试读结束:需要全本请在线购买: www.ertongbook.com

```
void main()
{...}
```

是一个函数。其中 main 是个函数名。每个 C++ 程序必须有且只能有一个 main()函数。所有 C++ 程序都从 main()函数开始执行,程序中的其他操作都是由 main()函数调用或激活的。void 说明该函数的类型。void 表示该函数执行后没有任何返回值。main()函数的缺省类型为整型,即

```
int main()
```

花括号{…}之间的内容为函数体。函数所需要的语句序列或过程都在花括号之间。花括号可以不单独占一行。

④函数体内的 cout 是标准输出,它表示的标准输出设备一般是屏幕。“ ≪ ”是个输出流插入运算符。与 cout 连用时,它表示把“ ≪ ”右边的内容在屏幕上显示出来。因为显示的内容是字符串,所以需要用双引号将“Hello,you are welcome!”括起来。

⑤第 4 行字符串"Hello,you are welcome!"后有一分号“;”,它是一个语句的结束标志。在 C++ 源程序中,一行可以写若干条语句,一个语句也可写在多行上,每个语句后必须用一个分号作为语句的结束。

⑥在 C++ 程序中可以从每一行的任意位置开始书写语句,不影响程序的执行。

例 1.2　编写程序从键盘任意输入两个数,输出这两个数的和。

```
#include <iostream.h>
void main()
{   int a,b,n;
    cout ≪"请输入两个数:";
    cin ≫ a ≫ b;
    n = a + b;
    cout ≪"a + b = " ≪ n ≪ endl;
}
```

程序说明如下。

①程序第 3 行的“int a,b,n;”为变量定义,定义 a、b、n 为整型变量名。在 C++ 程序中使用的变量都要在使用前定义。

②第 4 行“cout ≪"请输入两个数:";”是输出语句,在这里起到提示作用。

③第 5 行“cin ≫ a ≫ b;”中的 cin 代表标准输入设备,一般指键盘。“ ≫ ”是提取运算符。当它和 cin 联用时,接收用户从标准输入设备上输入的数据,并把接收到的数据赋给“ ≫ ”右边的变量。在一个 cin 语句中,“ ≫ ”可以连续使用多个。本例表示将从键盘上输入的第 1 个数送给变量 a,第 2 个数送给变量 b。从键盘上输入数据时,以空格、制表符或回车作为数据的分隔符。

④第 6 行是赋值运算,表示将从键盘上输入的 a 和 b 相加,结果存放在变量 n 中。

⑤第 7 行“a + b = ”原样输出,接着将变量名 n 中的内容在输出设备上输出。第 7 行中的 endl 为换行符,用来开始一个新行。若试图将下一个内容在新一行输出,可以使用 endl。endl 也可以用“'\n'”代替,效果一样。例如:

```
cout ≪"a + b = " ≪ n < '\n';
```

例1.2的执行示例如下：

```
请输入两个数:123  456
    a+b=579
```

例1.2程序也可以写成如下形式：

```
#include <iostream.h>
void main()
{   int a,b;
    cout<<"请输入两个数:"; cin>>a>>b;
    int n=a+b;
    cout<<"a+b="<<n;
    cout<<endl;
}
```

这个程序的运行结果与例1.2完全相同。注意这个程序中变量n的说明位置。C++允许根据需要随时说明新变量。

例1.3 编写由两个函数组成的C++程序。

```
#include <iostream.h>
int fmax(int a, int b)
{   if(a>b) return a;
    else return b;
}
int main()
{   int v1, v2;
    cout<<"请输入两个数:"; cin>>v1>>v2;
    cout<<"两个数中较大数是:"<<fmax(v1, v2)<<endl;
    return 0;
}
```

该程序由两个函数组成，即main()函数和fmax()函数。在主函数main()的cout中调用前面定义的fmax()函数，将fmax()函数执行后返回的结果在主函数的cout中显示输出。一个C++程序可以由若干个函数组成，程序由main()开始执行，一般结束也是在main()。注意，该程序中的main()函数为整型，故在程序结束前返回一个整数0。有关函数调用的详细内容将在第4章中介绍。

学习程序设计的过程总是由模仿开始，逐步深入，最后才能创造性地设计出自己的程序。这里的关键是要亲自动手编写程序，并且一定要上机调试、运行程序，从中发现问题、积累经验，使程序设计能力不断提高。

1.4 基本数据类型

数据是程序处理的对象。根据数据的特点，可将数据分为不同的类型。类型不同，存储方式不同，使用的场合也不同。程序中使用的数据必须属于某种数据类型。C++数据类型可以

分为基本数据类型和非基本数据类型，如图 1.2 所示。

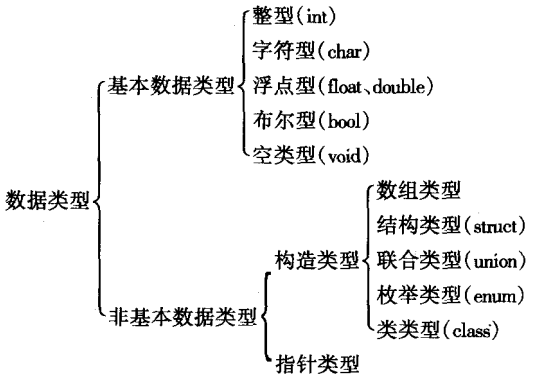

图 1.2　C++ 数据类型

基本数据类型是 C++ 系统已定义的类型。可以直接利用这些类型名来定义数据。表 1.1 给出 C++ 基本数据类型的名称以及在计算机内所占的位数(以字节为单位)和数据的取值范围。

表 1.1　C++ 基本数据类型

类型名	说明	字节	取值范围
bool	布尔型	1	true,false
[signed] char	有符号字符型	1	-128 ~ +127
unsigned char	无符号字符型	1	0 ~ 255
[signed] short int	有符号短整型	2	-32 768 ~ 32 767
unsigned short int	无符号短整型	2	0 ~ 65 535
[signed] int	有符号整型	4	-2 147 483 648 ~ +2 147 483 647
unsigned int	无符号整型	4	0 ~ 4 294 967 295
[signed]long int	长整型	4	-2 147 483 648 ~ +2 147 483 647
unsigned long int	无符号长整型	4	0 ~ 4 294 967 295
float	浮点型	4	$\pm(3.4\times10^{-38}\sim3.4\times10^{+38})$
double	双双精度型	8	$\pm(1.7\times10^{-308}\sim1.7\times10^{+308})$
long double	长双精度型	10	$\pm(1.2\times10^{-4\,932}\sim1.2\times10^{+4\,932})$

对表 1.1 说明如下。

①关键字 signed 和 unsigned 以及 short、long 被称为类型修饰符。使用 signed 修饰的数据类型称为有符号类型，这种类型的数据可以取正数、负数和 0，用[]括起来的 signed 可省略。使用 unsigned 修饰的数据类型称为无符号类型，这种类型的数据仅能取正数和 0。

②用 short 修饰的数据类型的值为 int 类型所占空间的一半或等于 int 类型的值域，但目前绝大多数编译系统都是等于 int 类型的值域。用 long 修饰的 int 数据类型的值大于或等于 int 类型的值域，具体值域的大小取决于具体的机器系统和所用的 C++ 编译系统。例如，在 32 位微机的 C++ 编译系统中，int 和 long 类型均占用 32 位，在 64 位机的 C++ 编译系统中它们均占用 64 位。

③用 short、long、unsigned 修饰 int 时可以省略 int。

④在 ISO C++ 标准中，float 为 7 位有效数字，double 为 15 位有效数字，long double 为 19 位

有效数字。在实际使用中各系统会有所不同。

1.5 常量、变量及引用

1.5.1 常量

程序中可以直接使用的常数称为常量。常量分为整型、浮点型、字符型以及字符串常量和布尔常量。

1.整型常量

(1)C++可以处理不同进制的整型常量

1)十进制整数　由0~9数字组成的正负整数,如0、15、-247。

2)八进制整数　以数字0开头的整数为八进制数。八进制数由数字0到7组成,如015、0236。

3)十六进制整数　以0x或0X开头的整数为十六进制数。十六进制数由数字0~9和字母a~f(或大写A~F)组成,如0x516、0x8AB、0xb2ff。

八进制和十六进制只能表示无符号整数,若写成-015或-0x8AB,系统并不能将其理解为负数。

(2)C++可以处理长整型量和无符号整型量

当任一整型常数后跟字母L(或l)时,表示是长整型量。若任一整型常数后跟字母u时,为无符号整型数,如12345L、7895u。

2.浮点型常量

浮点型又称实型,它由整数部分和小数部分组成。浮点型只有十进制形式。浮点型常数有两种表示形式。

1)小数形式　它由整数部分和小数部分组成,如3.14159、-0.55、-123.0。

2)指数形式　如+5.25e-8、0.5678e+05。其中 +5.25e-8表示$+5.25\times10^{-8}$,0.5678e+05表示0.5678×10^{5},字母e也可以大写。浮点型数中的"+"号可以省略。

指数形式表示浮点型数时,e(E)前可以是整数或小数,但e后的指数部分必须是整型数。

浮点型数总是按double类型存储的,只有在数的后面加上f才按float类型存储,如1.234E-6f。长双精度(long double)型常量通常在双精度数后面加上l或L表示,如1.2345e-12L。

3.字符型常量

字符型常量是用单引号括起来的单个字符,如'A'、'S'、'*'、'a'等。字符型常量说明如下。

①字符型常量中的单引号只作为定界符,不是字符型常量内容。

②字符型常量具有数值,其值就是该字符的ASCII码值。在C++程序中,字符型常量值可以作为整数参与运算,如'a'+5表示将'a'字符的ASCII码值与整数5相加,结果为字符'f'的ASCII码值(整数)。又如'9'-6表示数字字符9的ASCII码值减去整数6,结果为数字字符'3'的ASCII码值。又如'A'+32结果为'a','f'-'d'结果为整数2。

③字符型常量可以是ASCII字符集中任意可打印的字符,包括空格字符。

④字符型常量中另一种字符称为转义字符。转义字符用于表示ASCII字符集中控制代码或某些其他功能代码。转义字符由一个"\"开始,后跟一个字符,或一个"\"后跟一个八进制或

十六进制数,用单引号括起来。如'\n'表示回车换行,'\"'表示双引号。C++ 程序中的转义字符见表 1.2。

表 1.2　转义字符

字符形式	功能	字符形式	功能
\a	报警	\'	单引号
\b	退格	\"	双引号
\f	走纸(用于打印机)	\\	反斜杠
\n	换行	\0	空字符
\t	横向跳格(Tab 键)	\ddd	1到 3 位八进制数所表示的字符
\v	垂直跳格	\xhh	1到 2 位十六进制数所表示的字符
\r	回车		

表 1.2 中最后两行是用八进制或十六进制 ASCII 码表示的一个字符。例如,字符 a 的八进制 ASCII 码是 141,a 的十六进制 ASCII 码是 61,于是字符 a 可以表示成 '\141'或表示为 '\x61'。换行可以用'\012'或'\x0A'表示。

4.字符串常量

用双引号括起来的一串字符称为字符串,如"This is a string"、"a"、 "ABC　xyz\n"、"1234"。

字符串中可以包含空格、转义字符等任意字符,也可以是中文字符。双引号是字符串的定界符,计算字符串长度时双引号不计算在内。

在 C++ 程序中字符串常量的允许长度与所用环境有关,如 VC++6.0 为 2048。编译程序在存储字符串常量时自动在字符串最后加一个'\0'作为一个字符串的结束标志。'\0'占一个字节位置。因此计算字符串存储长度时总比用户写的实际字符串的字符个数多 1。例如字符串"This is a string" 的存储长度为 17,又如'A'和"A",其中'A'为字符常量,而"A"为字符串常量。前者存储长度为 1,后者存储长度为 2。

在计算机中一个字符占一个字节,而一个汉字占两个字节。

在程序设计中,字符常量用字符变量存放,而字符串通常用字符数组存放或用字符指针指向。

5.布尔常量

布尔常量仅有两个值,即 true(真)和 false(假)。

6.符号常量

除了前边介绍的在程序中直接使用的常量外,也可以为常量起一个名字,称为符号常量。符号常量在使用前必须进行说明。符号常量的说明形式为:

```
const 数据类型名 常量名 = 常量值;
```

或

```
数据类型名 const 常量名 = 常量值;
```

其中,const 为关键字;数据类型名为表 1.2 中的类型名。例如:

```
const int m = 100;
const float pi = 3.14159;
```

表示为整数 100 起名为 m,为浮点型数 3.14159 起名为 pi。说明后,在程序中当用到常数 100

或 3.14159 时,可以直接写它的符号名而不必再写该常数。

使用符号常量的好处:一是可以提高程序的可读性;二是增强程序的可维护性。如在程序中多次用到同一个常量,要修改该常量值只要修改该符号说明中的常量值即可,而不必逐一对程序中的常量值进行修改。使用符号常量不但提高了效率,也可避免由于疏忽而引起的错误。

使用符号常量时的注意事项如下:

①符号常量在说明时一定要赋初值,而且在程序中不能再对该符号常量重新赋值;

②符号常量名不要和一般变量名重名。

1.5.2 变量

在程序中可以改变值的量称为变量。

1.标识符

标识符用来为变量、符号常量、数组、函数、类型等命名。命名标识符的规则为:必须由字母、下画线和数字组成,而且第一个字符应是字母或下画线字符。例如,a、x1、data5、count 等为合法的标识符。标识符的长度视具体的编译系统而定。注意不要使用 C++ 的关键字(见附录 A)作为标识符。例如,int、for、static 等不能作为标识符。

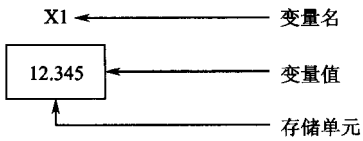

图 1.3 变量

变量名是标识符的一种。程序中使用的变量都要有一个变量名。在 C++ 中变量名区分大小写,因此 ex1 和 EX1 是两个不同的变量名。每个变量在内存中占有一定的存储单元,该存储单元中存放变量的值,其关系见图 1.3。程序中的变量名代表内存中的一个存储单元,每个存储单元都有一个地址。在程序设计中可以根据需要改变变量的值。

2.定义变量

在使用之前必须定义程序中的变量名。定义变量名时用表 1.1 中的类型名。例如:

```
char a,b,c;                 //定义 a、b 为字符型变量
int   x,y;                  //定义 x、y 为整型变量
long int s1,s2;             //定义 s1、s2 为长整型变量
float data1, data2;         //定义 data1、data2 为浮点型变量
double   w1,w2              //定义 w1、w2 为双精度型变量
unsigned m, n;              //定义 m、n 为无符号整型变量
```

3.变量的初始化

在定义变量的同时可以赋值,称为变量的初始化。例如:

```
char a = 'A';           //定义字符型变量 a 同时给 a 赋'A'字符
int   x = 0,y = 12 ;    //定义 x、y 为整型变量,同时给 x 赋初值 0,给 y 赋初值 12
double   w1 = 12.3456, w2 = -0.4567e-4;//定义 w1、w2 为双精度型变量并赋初值
```

变量的初始化也可以写成以下形式:

```
char a('A');
int x(0),y(12);
double w1(12.3456), w2(-0.4567e-4);
```

试读结束:需要全本请在线购买: www.ertongbook.com