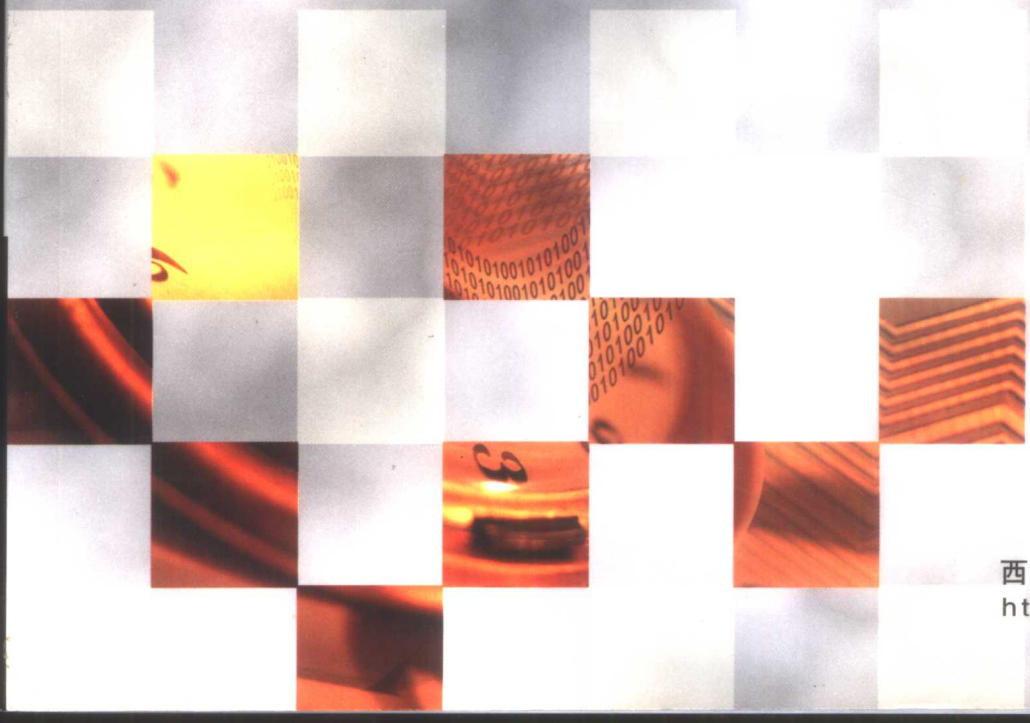
普通高等院校计算机类专业系列教材

The C++ Programming Language

C++程序设计语言

主编　揣锦华　副主编　李军民



西安电子科技大学出版社
http://www.xduph.com

普通高等院校计算机类专业系列教材

C++程序设计语言

主　编　揣锦华
副主编　李军民
主　审　朱战立

西安电子科技大学出版社
2003

内 容 简 介

C++是一种高效实用的程序设计语言，用它既可以进行结构化程序设计，又可以进行面向对象程序设计。本书将C++作为学习程序设计语言的入门语言，在系统介绍C语言本身的基础上，介绍了常用的数据结构和算法。本书依据作者多年教学经验写成，书中列举有大量实例，将复杂的概念用简洁浅显的语言来描述，力求深入浅出。本书配有相应的《C++程序设计语言经典题解与实验指导》，其中除了收集大量习题之外，还根据教学大纲，为每章配备有相应的上机实验内容，既方便教师安排教学，又便于读者上机实习。

本书适合作为高等学校计算机专业和非计算机专业程序设计课程的教材，也可供自学者使用。

为方便教学，本书配有电子教案，任课教师可与西安电子科技大学出版社联系，免费索取。

图书在版编目（CIP）数据

C++程序设计语言 / 揣锦华主编. —西安：西安电子科技大学出版社，2003.2

(普通高等院校计算机类专业系列教材)

ISBN 7-5606-1195-8

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

中国版本图书馆CIP数据核字（2002）第104130号

责任编辑 云立实

出版发行 西安电子科技大学出版社（西安市太白南路2号）

电 话 (029)8227828 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安文化彩印厂

版 次 2003年2月第1版 2003年6月第2次印刷

开 本 787毫米×1092毫米 1/16 印张 19.375

字 数 458千字

印 数 4 001～10 000册

定 价 20.00元

ISBN 7-5606-1195-8 / TP·0623 (课)

XDUP 1466001-2

*** * * 如有印装问题可调换 * * ***

前　言

随着计算机科学的发展，计算机技术已渗透到各学科的研究和应用之中，C语言被广泛地应用于各专业的科研开发。C++语言是从C语言发展演变而来的一种面向对象的程序设计语言。C++全面兼容了C，同时提供了比C更严格、更安全的语法。从这个意义上讲，C++首先是一个更好的C。

面向对象的程序设计(OOP)方法将数据及对数据的操作方法封装在一起，作为一个相互依存、不可分离的整体——对象。对同类型对象抽象出其共性，形成类。类中的大多数数据只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系，对象与对象之间通过消息进行通信。这样，程序模块间的关系简单，程序模块的独立性、数据的安全性具有良好的保障。同时，通过继承与多态性，使程序具有很高的可重用性，软件的开发和维护都更为方便。由于面向对象方法的突出优点，目前它已经成为开发大型软件时所采用的主要方法，而C++语言是面向对象的程序设计语言中应用最广泛的一种。

面向对象方法的出现，实际上是程序设计方法发展的一个返璞归真的过程。从本质上讲，软件开发就是对软件所要处理的问题域进行正确的认识，并把这种认识正确地描述出来。面向对象方法所强调的基本原则就是直接面对客观存在的事物来进行软件开发，将人们在日常生活中习惯的思维方式和表达方式应用在软件开发中，使软件开发从过分专业化的方法、规则和技巧中回到客观世界，回到人们通常的思维。

虽然C++语言是从C语言发展而来的，但是C++本身也是一个完整的程序设计语言，而且它与C语言的程序设计思想完全不同。因此，我们认为C++语言可以作为程序设计的入门语言来学习。

本书将C++作为学习程序设计语言的入门语言，不仅详细介绍了语言本身，而且还介绍了常用的数据结构和算法。全书以面向对象的程序设计方法贯穿始终，从面向对象的理论到面向对象的实现，力求使读者在掌握基本程序设计方法的同时，牢固树立面向对象的编程思想，为其它后续课程的学习打下坚实的基础，以适应当前软件发展的需要。本书在讲解语法时，着重从程序设计方法学的角度讲述其意义和用途。本书的宗旨是，不仅要使读者掌握C++语言本身，而且能够对现实世界中较简单的问题及其解决方法用面向对象的语言进行描述。当然，要达到能够描述较复杂问题域的水平，还需要学习其它相关的课程。

针对初学者的特点，本书力求做到深入浅出，将复杂的概念用简洁浅显的语言来讲述，使读者可以轻松地入门，循序渐进地提高。

本书依据作者多年教学经验写成，对每一部分的知识点和难点，都力求以比较精练的语言进行讲解。同时，对每一个知识点都列举了必要的例题，并对例题进行了比较深刻的分析。为了帮助读者更好地使用本教材，本书配有相应的《C++程序设计语言经典题解与实验指导》，其中除了收集有大量典型例题和习题之外，还为每章配备有相应的上机实验内容，教师可以依此安排教学和上机实验。

本书适合作为高等学校计算机专业和非计算机专业程序设计课程的教材，也可供自学者使用。本书的建议教学学时为40学时，上机实验学时为20学时。

本书的第1章由长安大学的郭兰英编写，第2章由长安大学的汤娟编写，第3章和第6章由西安科技学院的李军民编写，第4章由西安工程科技学院的牟莉编写，第9章由长安大学的赵彦峰编写，第10章由长安大学的许宏科编写，第5章、第7章、第8章和第11章由长安大学的揣锦华编写。西安石油学院的朱战立老师为本书的审稿人。在本书的编写过程中，查阅和参考了部分文献，在此对书后所列出的参考文献的作者表示感谢。由于作者水平有限，书中难免有不足和错误，恳请读者批评指正。

编　者

2002年10月

普通高等学校计算机类专业系列教材

编审专家委员会名单

主 任 委 员：**冯博琴**　（陕西省计算机教育学会理事长，
西安交通大学计算机教学实验中心主任，教授）

副主任委员：**陈建铎**　（陕西省计算机教育学会副理事长，
西安石油学院计算机系教授）

李伟华　（陕西省计算机教育学会副理事长，
西北工业大学计算机系副主任，教授）

武　波　（陕西省计算机教育学会副理事长，
西安电子科技大学计算机学院副院长，教授）

李荣才　（西安电子科技大学出版社总编辑，教授）

委员：（按姓氏笔划排列）

巨永锋　（长安大学信息工程学院副院长，副教授）
冯德民　（陕西师范大学计算机科学学院院长，教授）
石美红　（西安工程科技学院信息控制系副教授）
朱明放　（陕西理工学院计算机系副主任，副教授）
何东健　（西北农林科技大学信息工程学院院长，教授）
陈　桦　（陕西科技大学计算机与信息科学系主任，教授）
李长河　（西安理工大学计算机科学与工程系主任，副教授）
李晋惠　（西安工业学院计算机系副主任，副教授）
李银兴　（宝鸡文理学院计算机系副主任，副教授）
张俊兰　（延安大学计算机系教授）
孟东升　（西安石油学院计算机系副主任，副教授）
赵文静　（西安建筑科技大学信息与控制工程学院副院长，教授）
耿国华　（西北大学软件开发中心主任，教授）
龚尚福　（西安科技学院计算机系主任，教授）

项目策划　陈宇光　马乐惠
策　　划　云立实　马武装　臧延新　马晓娟
电子教案　马武装

目　　录

第 1 章　C++简单程序设计

本章要点

- 最简单的 C++程序
- C++的数据类型及表达式
- 程序的基本控制结构

程序设计工作主要包括数据结构（即数据类型）和算法（操作步骤）的设计。程序中最基本的元素是数据类型。确定了数据类型，才能确定变量空间的大小和其上的操作。C++的数据类型检查与控制机制，奠定了 C++ 今天的地位。本章主要介绍 C++的基本数据类型和自定义数据类型。算法由一系列控制结构组成，本章介绍的顺序、选择和循环结构是程序设计中最基本的控制结构，也是构成复杂算法的基础。

1.1　C++ 语言概述

1.1.1　从 C 到 C++

C++语言是从 C 语言发展演变而来的，因此在介绍 C++语言之前，我们首先介绍一下 C 语言。C 语言最初是美国贝尔实验室的戴尼斯·M·利奇（Dennis. M. Ritchie）在 B 语言基础上开发出来的，并于 1972 年在一台 PDP－11 计算机上实现了最初的 C 语言。目前，比较流行的 C 语言版本基本上都是以 ANSI C 为基础的。

C 语言具有以下优点：

① 语言简洁灵活。

② 运算符和数据结构丰富，具有结构化控制语句，程序执行效率高。

③ 与高级语言相比，具有可以直接访问物理地址，能进行位运算的优点。

④ 与汇编语言相比，又具有良好的可读性和可移植性。

尽管如此，C 语言毕竟是一个面向过程的编程语言，因此与其它面向过程的编程语言一样，已经不能满足运用面向对象方法开发软件的需要。C++语言便是在 C 语言基础上为支持面向对象的程序设计而研制的一个通用的程序设计语言，它是在 1980 年由贝尔实验室的 Bjarne Stroustrup 博士创建的。

C++包含了整个 C，C 是建立 C++的基础。C++包括 C 的全部特征、属性和优点，同时，C++添加了对面向对象编程的完全支持。

1.1.2 一个简单的C++程序

现在，我们来看一个简单的程序实例。例1-1是一个面向过程的程序，我们只是通过这个程序看一看，计算机程序是个什么样子，人们如何通过程序来控制计算机的操作。

【例1-1】 一个简单的C++程序例题。

```
#include<iostream.h>
void main( )
{
    cout<<"hello!\n";
    cout<<"I am a student.\n";
}
```

这里main是主函数名，函数体用一对大括号括住。C++程序由函数构成。在C++程序中，必须有且只能有一个名为main( )的函数，因为程序总是从main( )开始执行的。main( )函数之前的void表示main( )函数没有返回值（关于函数的返回值将在第2章介绍）。程序由语句组成，每条语句由分号“；”作为结束符。cout是一个输出流对象，它是C++系统预定义的对象，其中包含了许多有用的输出功能。输出操作由操作符“<<”来表达，其功能是将紧随其后的双引号中的字符串输出到标准输出设备（显示器）上。在第9章中将对输出流做详细介绍，在这里，读者只要知道这段程序可以实现在显示器上输出

```
hello!
I am a student.
```

就可以了。

程序中的

#include <iostream.h>

的作用是在编译之前，将文件iostream.h中的代码嵌入到程序中该指令所在的地方。作为程序的一部分，iostream.h文件中声明了程序所需要的输入和输出操作的有关信息。cout和“<<”操作的有关信息就是在该文件中声明的。由于这类文件常被嵌入在程序的开始处，所以称之为头文件。在C++程序中如果使用了系统中提供的一些功能，就必须嵌入相关的头文件。

当我们编写完程序文本后，要将它存储为后缀为.cpp的文件，此文件称为C++源文件，再经过编译系统的编译、连接后，最后产生出后缀为.exe的可执行文件。

通过以上程序可以看出，C++的程序结构由编译预处理、程序主体和注释组成，其特点如下：

① 每个以符号“#”开头的行，称为编译预处理语句。编译预处理是C++组织程序的工具。有关#include语句的作用及其使用方法，将在第5章进行详细介绍。

② 一个C++程序可以由一个或多个函数组成。任何一个完整的C++程序，都必须包含一个且只能包含一个名为main( )的函数，程序总是从main( )函数开始执行，而不管main( )函数处于程序的什么位置。

③ 函数体应由“{}”括起来。函数体一般包括变量的定义部分和执行部分。所有的变量要先定义后使用。

④ 注释是程序员为读者写的说明，是提高程序可读性的一种手段。一般可将注释分为两种：序言注释和注解注释。前者用于程序开头，说明程序或文件的名称、用途、编写时间、编写人以及输入、输出等，后者用于程序难懂的地方。

C++的注释为“//”之后的内容，直到换行。注释仅供阅读程序使用，是程序的可选部分。在生成可执行程序之前，C++忽略注释，并把每个注释都视为一个空格。另外，C++还兼容了 C 语言的注释。

⑤ 每个语句和数据定义的后面都要有一个分号。这一点初学者尤其要注意。

⑥ main 函数名和关键字（如 void、int、float 等）都是小写字母构成。C++程序中的标识符是大小写“敏感”的，所以，在书写标识符的时候要注意其大小写。

1.1.3　字符集

C++语言的字符集由下述字符构成:

① 英文字母：A～Z，a～z

② 数字字符：0～9

③ 特殊字符：空格　!　#　%　^　&　*　_（下划线）

+　=　:　-　～　<　>　/　\

`　"　;　.　()　{}　[]

1.1.4　词法记号

词法(语句)记号是构成语句的最小单元，这里我们介绍 C++的标识符、关键字、文字、运算符、分隔符和空白符。

1. 标识符

标识符是程序员声明的字符序列，它命名程序正文中的一些实体，如函数名、变量名、类名、对象名等。C++标识符的构成规则如下：

① 以大写字母、小写字母或下划线(_)开始。

② 可以由大写字母、小写字母、下划线(_)或数字 0～9 组成。

③ 大写字母和小写字母代表不同的标识符。

④ 不能是 C++关键字。

例如，Richad、red_line、_No1 都是合法的标识符，而 No.1、1^{st} 则是不合法的标识符。

在标识符的命名中要特别注意：C++是大小写敏感的，即大写和小写字母被认为是不同的字母。例如，标识符 something、Something、SOMETHING、SomeThing 都被视为不同的名字。

2. 关键字

关键字是 C++预定义好的标识符，这些标识符对 C++编译系统有着特殊的含义。如

例 1-1 中的 void，后面将逐步介绍到的数据定义语句 int、float、long 及 double 等等，它们都是 C++的关键字。

3. 文字

文字是在程序中直接使用符号表示的数据，包括数字、字符、字符串和布尔文字。在本章 1.2 节中将详细介绍各种文字。

4. 操作符(运算符)

操作符是用于实现各种运算的符号，例如＋、－、*、／等。在本章 1.2 节及后续章节中将详细介绍各种操作符。

5. 分隔符

分隔符用于分隔各个词法记号或程序正文。C++的分隔符如下：

() {} , : ;

这些分隔符不表示任何实际的操作，仅用于构造程序。例如，“{ }”用于分隔函数体，“；”作为语句的分隔符，其它分隔符的具体用法会在以后相应的章节中介绍。

6. 空白符

在程序编译时的词法分析阶段将程序正文分解为词法记号和空白。空白是空格、制表符(TAB 键产生的字符)、换行符(Enter 键所产生的字符)和注释的总称。

空白符用于指示词法记号的开始和结束位置，但除了这一功能之外，其余的空白将被忽略。因此，C++程序可以不必严格地按行书写，凡是可以出现空格的地方，都可以出现换行。例如：

```
int  j;
```

与

```
int     j;
```

或与

```
int
j
;
```

是等价的。但是尽管如此，我们在书写程序时，仍要力求清晰、易读。因为一个程序不仅要让机器执行，还要让人阅读的同时便于修改、维护。

1.2 基本数据类型和表达式

程序处理的对象是数据。数据有许多种类，例如数值数据、文字数据、图像数据以及声音数据等，但其中最基本也是最常用的是数值数据和文字数据。

无论什么数据，在对其进行处理时都要先存放在内存中。显然，不同类型的数据在内存中的存放格式也不相同，甚至同一类数据，为了处理方便，也可以使用不同的存储格式。例如，数值数据其存储格式又可以分为整型、长整型、浮点型和双精度型等几种类型；文字数据也可以分为单个字符和字符串。因此，程序中在对各种数据进行处理之前都要对其

类型预先加以说明。这样做一是便于为这些数据分配相应的存储空间，二是说明了程序处理数据时应采用何种运算方法。因为不同类型的数据有不同的运算方法。例如，整数和实数可以参加算术运算；字符串可以拼接；逻辑数据可以参加“与”、“或”、“非”等逻辑运算。

我们编写计算机程序的目的，就是为了解决客观世界中的现实问题。所以，高级语言中提供了丰富的数据类型和运算符。C++中的数据类型分为基本类型和非基本类型，见图 1-1。基本类型是 C++编译系统内置的，非基本类型也称为用户定义数据类型，即用户自己定义的数据类型，本节我们首先介绍基本数据类型。

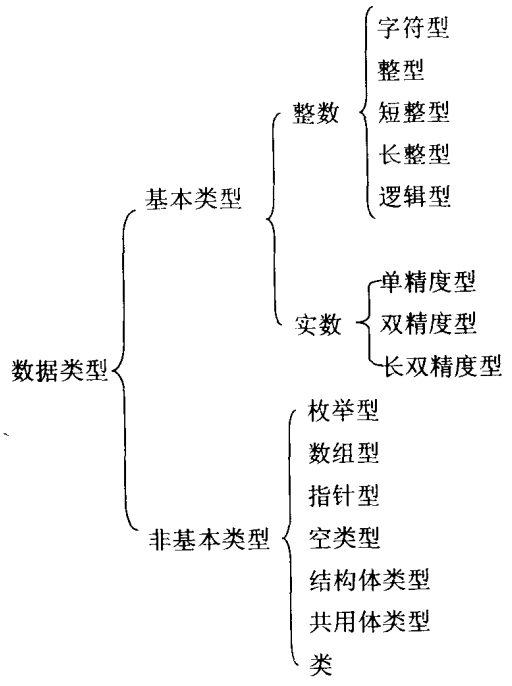


图 1-1　C++ 的数据类型

1.2.1　基本数据类型

一个程序要运行，就要先描述算法。描述一个算法应先说明算法要用的数据，数据以变量或常量的形式来描述。每个变量或常量都有数据类型。

变量是存储信息的单元，它对应于某个内存空间。为了便于描述，计算机高级语言中都用变量名来表示其内存空间，所以，程序能在变量中存储值和取出值。

在定义变量时，说明变量名和数据类型(如 int、float)就是告诉编译器要为变量分配多少空间，以及变量中要存储什么类型的值。数据类型的定义确定了其内存所占空间大小，也确定了其表示范围。表 1-1 列出了基本数据类型的取值范围。

在不同的系统中，每个变量类型所占的字节数目可能有所不同，表 1-1 里列出的是在 VC++6.0 编译环境中的情况（也是目前大多数编译环境中的情况）。

表 1-1　常用基本数据类型描述

类　型	说　明	长度	表　示　范　围	备　注
bool	逻辑型	1	false,true	
char	字符型	1	-128～127	-2^7～(2^7-1)
unsigned char	无符号字符型	1	0～255	0～(2^8-1)
short	短整形	2	-32768～32767	-2^{15}～$(2^{15}-1)$
unsigned short	无符号短整型	2	0～65535	0～$(2^{16}-1)$
int	整型	4	-2147483648～2147483647	-2^{31}～$(2^{31}-1)$
unsigned int	无符号整型	4	0～4294967295	0～$(2^{32}-1)$
long	长整型	4	-2147483648～2147483647	-2^{31}～$(2^{31}-1)$
unsigned long	无符号长整型	4	0～4294967295	0～$(2^{32}-1)$
float	浮点型	4	-3.4×10^{38}～3.4×10^{38}	7 位有效位
double	双精度	8	-1.7×10^{308}～1.7×10^{308}	15 位有效位
long double	长双精度	8	-1.7×10^{308}～1.7×10^{308}	15 位有效位

在大多数系统中，short int 表示两个字节。short 只能修饰 int，short int 可省略为 short。long 只能修饰 int 和 double，long int 可简称为 long。unsigned 和 signed 只能修饰 char 和 int。一般情况下，默认的 char 和 int 为 signed。实型 float 和 double 总是有符号的，不能用 unsigned 修饰。

1.2.2　常量

所谓常量，是指在程序运行的整个过程中其值始终不可改变的量。例如，68、3.5、'A'、"hello!" 都是常量。常量有以下几种。

1．整型常量

整型常量即以数码形式出现的整数，包括正整数、负整数和零。整型常量的表示形式有十进制、八进制和十六进制。

十进制整型常量的一般形式与数学中我们所熟悉的表示形式是一样的：

[±] 若干个 0～9 的数字

即符号加若干个 0～9 的数字，但数字部分不能以 0 开头，正数前边的正号可以省略。

八进制整型常量的数字部分要以数字 0 开头，一般形式为

[±] 0 若干个 0～7 的数字

十六进制整型常量的数字部分要以 0x 开头，一般形式为

[±] 0x 若干个 0～9 的数字及 A～F 的字母(大小写均可)

整型常量可以用后缀字母 L（或 l）表示长整型，用后缀字母 U（或 u）表示无符号型，也可同时使用后缀 L 和 U（大小写无关）。

例如，123、0123、－0x5af、127L、127l 都是合法的常量形式。

2．实型常量

实型常量又称浮点小数。在 C++语言中，实型常量只使用十进制表示，有两种表示形

式：一般形式和指数形式。

一般形式：例如，16.5、-13.5 等。

指数形式：例如，0.565E+2 表示 0.565×10^{2}，-34.4E-3 表示 -34.4×10^{-3}，其中，字母 E 可以大写或小写。当以指数形式表示一个实数时，整数部分和小数部分可以省略其一，但不能都省略。例如，.234E-1 和 12.E2 都是正确的，但不能写成 E-3 这种形式。

实型常量默认为 double 型，如果后缀为 F（或 f）则为 float 型。

3. 字符常量

字符常量是单引号括起来的一个字符，如 'a'、'G'、'?'、'$' 等。

另外，还有一些字符是不可显示字符，也无法通过键盘输入，例如响铃、换行、制表符、回车等等。这样的字符常量该如何写到程序中呢？C++提供了一种称为转义序列的表示方法来表示这些字符，表 1-2 列出了 C++预定义的转义序列。

表 1-2　C++预定义的转义序列

字符形式	ASCII 码(十六进制)	功　　能
\n	0A	换行
\t	09	横向跳格（即跳到下一个输出区）
\v	0B	竖向跳格
\b	08	退格
\r	0D	回车
\a	07	响铃
\\	5C	反斜杠字符“\”
\'	27	单引号
\"	22	双引号
\ddd	ddd(八进制)	1 到 3 位八进制数所代表的字符
\xhh	hh	1 到 2 位十六进制数所代表的字符

由于单引号是字符的界限符，所以单引号本身就要用转义序列表示为 '\'。同理，反斜杠是用来描述转义字符的，所以反斜杠本身也要用转义序列表示为 '\\'。像这样的字符还有双引号，因为双引号是字符串的界限符，所以显示双引号也要用转义序列表示为 '\" '。

例如，把例 1-1 中的 cout 语句改为

```
cout<<"\x07operating    \tsystem\n";
```

执行结果就是，响铃且输出字符串“operating system”。

字符数据在内存中以 ASCII 码的形式存储，每个字符占一个字节，使用七个二进制位。

4. 字符串常量

字符串常量简称字符串，是用一对双引号括起来的字符序列。例如，"China"、"1234" 都是字符串常量。字符串与字符是不同的，字符串在内存中的存放形式是：按串中字符的排列次序顺序存放对应字符的 ASCII 码，每个字符占一个字节，并在字符串末尾添加 '\0'作为结束标记。图 1-2 是字符数据及其存储形式（十六进制）的举例。从图 1-2 中可以看出，字符串 "a" 与字符 'a' 是不同的。

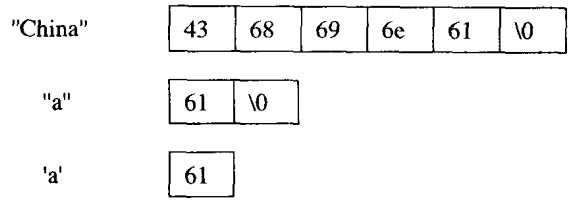


图 1-2　字符数据的存储形式

5. 布尔型常量

布尔型常量只有两个：false(假)和 true(真)。

1.2.3　变量

在程序的执行过程中其值可以变化的量称为变量，变量需要用标识符来命名。就像常量具有各种类型一样，变量也具有相应的类型。变量在使用之前需要首先声明其类型和名称。在同一语句中可以声明同一类型的多个变量。变量声明的形式如下：

<类型标识符>　变量名 1，变量名 2，…，变量名 n;

在程序运行时系统会给每一个声明过的变量分配内存空间，用于存放对应类型的数据，因而变量名也就是对相应内存单元的命名。在声明一个变量的同时，也可以给它赋以初值，而这实质上就是给对应的内存单元赋值。例如：

```
int a=3;
float f=3.45;
char c='b';
```

有一点值得注意，虽然 C++ 中有字符串常量，却没有字符串变量。那么，用什么类型的变量来存放字符串呢？在后面的章节，我们会介绍用字符数组来存储字符串常量。

1.2.4　引用

程序设计语言的进化使用户从被迫解决细节问题中解脱出来，而转向允许用户花更多的时间来考虑“大的蓝图”。根据这种精神，C++包含了一个称为引用的特性。本小节仅仅介绍了引用的基本概念，有关更详细的内容在后面章节会陆续介绍，到时读者就会掌握引用的使用方法。

引用是个别名，当建立引用时，程序用另一个变量或对象（目标）的名字来对其进行初始化。自此，引用作为目标的别名而使用，对引用的改动实际是对目标的改动。

引用的声明形式为

<类型标识符>　&引用名=目标名

或

<类型标识符> & 引用名=目标名

其中：

① 引用名是为引用型变量所起的名字，它必须遵循变量的命名规则。

② 前面的数据类型就是它所引用目标的数据类型。