

NCRE 研究组 编写

全国计算机等级考试

二级应试辅导

——C++语言程序设计



高等教育出版社

全国计算机等级考试

二级应试辅导 ——C++ 语言程序设计

NCRE 研究组 编写

高等教育出版社

内容提要

本书是与教育部考试中心组织编写的《全国计算机等级考试二级教程——C++ 语言程序设计》相配套的辅导书,各章内容与教程一一对应。

本书每章均包括学习目标与要求、内容要点、例题分析与解答、自测题和自测题答案五个部分。各章在内容要点的基础上补充了一些必要的内容和说明,以加强对要点的理解。全书通过大量例题分析和解答,帮助考生掌握 C++ 编程技术,并增加大量自测题和参考答案供考生练习和参考。

图书在版编目(CIP)数据

全国计算机等级考试二级应试辅导——C++ 语言程序设计 / NCRE 研究组编写. —北京:高等教育出版社,2005.3

ISBN 7-04-017140-6

I.全... II.N... III.①电子计算机—水平考试—自学参考资料②C语言—程序设计—水平考试—自学参考资料 IV.TP3

中国版本图书馆 CIP 数据核字(2005)第 013077 号

策划编辑	肖子东	责任编辑	肖子东	封面设计	于涛
责任绘图	吴文信	版式设计	马静如	责任校对	胡晓琪
责任印制	孔源				

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 北京蓝色畅想图书发行有限公司
印 刷 北京铭成印刷有限公司

开 本 850×1168 1/16
印 张 15
字 数 370 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>

版 次 2005 年 3 月第 1 版
印 次 2005 年 3 月第 1 次印刷
定 价 28.60 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 17140-00

前 言

全国计算机等级考试自 1994 年开考以来，适应了社会的需要，得到了社会的广泛认可，在推广和普及计算机应用知识和技能等方面发挥了重要作用。全国计算机等级考试是一种面向社会的、开放的、非学历的职业教育和继续教育形式。考试不是目的，而以考促学，为构建国家终身教育学习体系尽一份力量，才是全国计算机等级考试的最终目标。为了给广大考生提供更多的学习帮助和支持，高等教育出版社组织编写了这套全国计算机等级考试应试辅导丛书。

本书是与教育部考试中心编写的《全国计算机等级考试二级教程——C++ 语言程序设计》（在本书中简称“教程”）相配套的辅导书，各章的内容与教程相对应，体现了考试大纲的要求，既不超纲，又不降低水平，讲解简明扼要。本书各章均包括学习目标与要求、内容要点、例题分析与解答、自测题及答案五个部分。各章在概括主要内容要点的基础上，对大量的例题做了分析和解答，同时编制了大量的自测题供考生练习并给出了参考答案。

由于编写时间仓促，难免存在疏漏与不足之处，敬请广大读者批评指正。

编 者
2004 年 12 月

目 录

第 1 章	C++ 语言概述	1
1.1	学习目标与要求	1
1.2	内容要点	1
1.3	例题分析与解答	4
1.4	自测题	6
1.5	自测题答案	7
第 2 章	数据类型、运算符和表达式	8
2.1	学习目标与要求	8
2.2	内容要点	8
2.3	例题分析与解答	17
2.4	自测题	23
2.5	自测题答案	25
第 3 章	基本控制结构	26
3.1	学习目标与要求	26
3.2	内容要点	26
3.3	例题分析与解答	32
3.4	自测题	47
3.5	自测题答案	54
第 4 章	数组、指针与引用	57
4.1	学习目标与要求	57
4.2	内容要点	57
4.3	例题分析与解答	68
4.4	自测题	75
4.5	自测题答案	79
第 5 章	函数	81
5.1	学习目标与要求	81
5.2	内容要点	81
5.3	例题分析与解答	92
5.4	自测题	100
5.5	自测题答案	106

第 6 章	类和对象	108
6.1	学习目标与要求	108
6.2	内容要点	109
6.3	例题分析与解答	120
6.4	自测题	145
6.5	自测题答案	154
第 7 章	继承和派生	159
7.1	学习目标与要求	159
7.2	内容要点	159
7.3	例题分析与解答	167
7.4	自测题	174
7.5	自测题答案	179
第 8 章	运算符重载	180
8.1	学习目标与要求	180
8.2	内容要点	180
8.3	例题分析与解答	182
8.4	自测题	189
8.5	自测题答案	194
第 9 章	模板	195
9.1	学习目标与要求	195
9.2	内容要点	195
9.3	例题分析与解答	202
9.4	自测题	209
9.5	自测题答案	214
第 10 章	C++ 流	215
10.1	学习目标与要求	215
10.2	内容要点	215
10.3	例题分析与解答	221
10.4	自测题	227
10.5	自测题答案	231

第1章 C++ 语言概述

1.1 学习目标与要求

1. 了解 C++ 语言的特点和面向对象程序设计的基本概念。
2. 了解组成 C++ 语言的基本符号。
3. 了解 C++ 语言中的五种词汇：关键字、标识符、字面常量、运算符和标点符号。
4. 掌握 C++ 程序的两种基本框架：
 - (1) C++ 语言的结构化程序设计框架。
 - (2) C++ 语言的面向对象程序设计框架。
5. 掌握 C++ 程序的开发过程，包括编辑、编译、链接、运行和调试。

1.2 内容要点

一、C++ 程序设计语言

C++ 程序设计语言（又称 C++ 语言或 C++）是以 C 语言为基础发展起来的一种高级程序设计语言。C++ 语言有以下特点：

- (1) C++ 是一种面向对象的程序设计语言。
- (2) C++ 是 C 语言的超集。
- (3) C++ 是程序员和软件开发者在实践中创造的。

C++ 保持了对 C 语言的兼容性，继承了 C 语言的众多优点，同时提供了对面向对象程序设计的全面支持。

二、面向对象程序设计

面向对象方法的本质是：主张从客观世界固有的事物出发来构造系统，提倡使用人类自然的思维方法来认识、理解和描述客观事物，使得系统中的对象以及对象之间的关系能够如实地反映问题领域中的固有事物及其关系。

面向对象程序设计的基本概念包括：

1. 对象和类

对象是系统中用来描述客观事物的一个实体，它由一组表示其特征的属性和一组可以执行的操作构成。属性是描述对象性质的数据；操作是对象可以完成的功能。类就是对具有相同属性和

说明：

- (1) 程序中由用户命名的标识符不能与关键字重名。
- (2) 标识符是以字母或下画线开头的，由字母、数字、下画线组成的字符串。
- (3) 标识符中的字母区分大小写，myCar 和 MyCar 是不同的标识符。
- (4) 字面常量分为四类，即整型、浮点型、字符型和字符串型。
- (5) 有三个运算符是由关键字充当的，它们包括：sizeof、new、delete。

四、C++ 程序的两种基本框架

1. 结构化程序设计框架

既然 C++ 起源于 C 语言，则可将 C++ 当作 C 语言的改进版本，进行传统的结构化程序设计。在 C++ 的结构化程序设计框架中函数是程序的基本模块；使用的编程方法是：功能分解，逐步求精。某个程序中的函数调用关系和相应的实现代码可用图 1.2 表示。

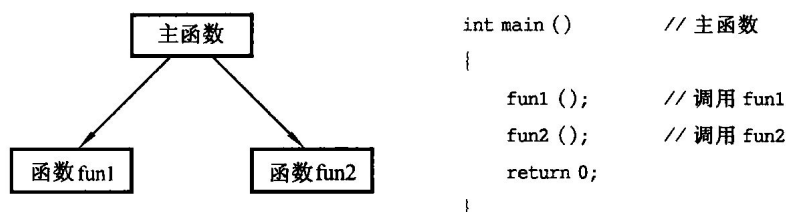


图 1.2 函数调用关系和相应的实现代码

2. 面向对象程序设计框架

作为重要的面向对象程序设计语言，C++ 对 OOP 方法提供了完整的支持。在 C++ 的面向对象程序设计框架中类是程序的基本模块，使用的编程方法是：分析类和对象，建立层次结构。某个程序中的类层次关系和相应的实现代码可用图 1.3 表示。

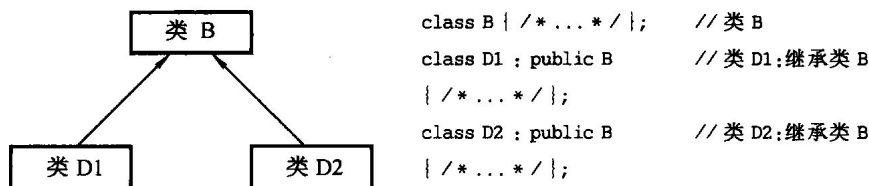


图 1.3 类层次关系和相应的实现代码

五、C++ 程序的开发过程

作为一种编译型高级语言，C++ 程序的开发步骤包括：

- (1) 编辑：使用文本编辑器编写 C++ 源程序，包括 .cpp 文件和 .h 文件；
- (2) 编译：使用 C++ 编译器将源程序文件翻译成目标代码，形成 .obj 文件；
- (3) 链接：使用链接器将 .obj 文件与所需库文件链接起来，形成 .exe 可执行文件；
- (4) 执行：操作系统会将 .exe 文件加载到内存中运行；
- (5) 调试：使用调试器对程序进行跟踪调试，以排除在执行过程中发现的错误。

由 C++ 源程序转换到可执行程序的过程好比将英语翻译成汉语，其中 C++ 编译器充当着“译员”的角色，如图 1.4 所示。

Microsoft Visual C++ 6.0（简称 VC6）是一个 C++ 编译系统，它提供了 C++ 项目的集成开发环境（Integrated Development Environment，简称 IDE），程序的所有开发步骤都可以在 VC6 中完成。有关使用 VC6 编写标准 C++ 程序的具体方法请参阅教程第 275 页附录 A。

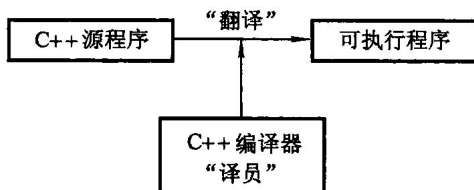


图 1.4 将 C++ 源程序转换为可执行程序

1.3 例题分析与解答

例 1 下列叙述中正确的是

- A) C++ 是 C 语言的子集
- B) 学习 C++ 之前必须先学习 C 语言
- C) C++ 与 C 语言都是面向对象程序设计语言
- D) C++ 继承了 C 语言简明、高效和灵活的特点

【答案】D

【分析】C++ 语言的最初设计宗旨就是在不改变 C 语言语法规则的基础上添加新特性，可以说 C++ 几乎包含 C 语言的全部语法特征，所以 C++ 是 C 语言的超集，而不是子集，故选项 A 错误。

由于 C++ 中的面向过程部分相当于功能增强的 C 语言，所以在学习 C++ 之前没有必要先学 C 语言，这也是 C++ 设计者 Bjarne Stroustrup 和大多数 C++ 程序员所持的观点，故选项 B 错误。

C 语言能够很好地支持结构化程序设计，但不能直接支持面向对象程序设计，所以 C 语言是一种结构化程序设计语言；C++ 提供对面向对象程序设计的全面支持，所以 C++ 是一种面向对象程序设计语言，故选项 C 错误。

能够很好地兼容 C 语言，继承其简明、高效、灵活等优点，正是 C++ 取得成功的原因之一，故选项 D 正确。

例 2 下列选项中属于对象的是

- A) 人类
- B) 一只叫“咪咪”的小花猫
- C) 气球充气
- D) 圆的半径

【答案】B

【分析】“人类”是一个抽象概念，是对具有相同属性和行为的个体的描述，所以“人类”是一个类，张三、李四、王五……这些具体的人才是“人类”的对象（或称为实例）。

选项 B 中的小花猫“咪咪”是指现实世界中存在的一只具体可见的动物，它是猫类的一个对象，故选项 B 正确。

“气球充气”是气球对象可以完成的功能，是气球类对外界提供的操作，因此“气球充气”是操作而不是对象。

“圆的半径”用来描述圆形的性质，它是圆形类的属性而不是具体的对象。

例 3 下列 4 组标识符中全部合法的一组是

- | | | | |
|----------|----------|-------------|---------|
| A) hello | B) _001 | C) longName | D) 3COM |
| 012 | u_name | _k2m | visual |
| var0 | pay@mail | IF | DEFINED |

【答案】C

【分析】C++ 中标识符的命名规则是：以字母或下画线开头，由字母、数字、下画线组成的字符串。注意：标识符不能与 C++ 关键字同名，标识符中的字母要区分大小写。

按照上述规则，选项 A 中的 012 以数字 0 开头，因此不是合法标识符；选项 B 中的 pay@mail 含有字符@，它不属于字母、数字或下画线，因此也不是合法标识符；选项 C 中的 IF 由大写字母组成，它并不是关键字 if (C++ 关键字中的字母均为小写)，所以是合法标识符；选项 D 中的 3COM 以数字 3 开头，因此不是合法标识符。

例 4 请将下面的 C++ 程序补充完整，使其中的函数调用关系为：主函数按顺序依次调用函数 fun1 和函数 fun2；在函数 fun2 中调用函数 fun3。（与函数调用无关的内容已省略。）

```
void fun1();
void fun2();
void fun3();
int main()
{
    [1];
    [2];
    return 0;
}

void fun1() { /* ... */ }
void fun2()
{
    [3];
}

void fun3() { /* ... */ }
```

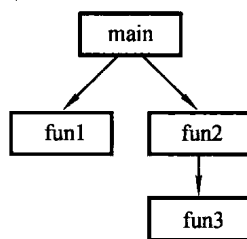


图 1.5 函数调用关系(例 4)

【答案】[1] fun1 () [2] fun2 () [3] fun3 ()

【分析】本程序主要体现了主函数 main 与函数 fun1、fun2 和 fun3 之间的调用关系，属于 C++ 结构化程序设计框架。按照题目要求，在主函数 main 中应先调用函数 fun1，再调用函数 fun2，且调用次序不能颠倒，故在 [1] 和 [2] 处分别填写函数调用表达式 fun1 () 和 fun2 ()。另外，在函数 fun2 中还要调用函数 fun3，故在 [3] 处填写函数调用表达式 fun3 ()。整个程序中的函数调用关系如图 1.5 所示。

例 5 请将下面的程序段补充完整，使其中的类层次关系为：类 C2 和类 C3 都继承自类 C1；类 C4 继承自类 C2 和类 C3。（与类层次关系无关的内容已省略。）

```

class C1 { /* ... */ };
class C2: public [1]
{ /* ... */ };
class C3: public [2]
{ /* ... */ };
class C4: public [3], public [4]
{ /* ... */ };

```

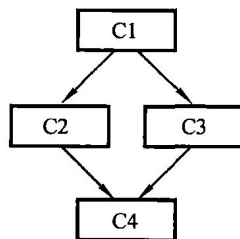


图 1.6 类层次关系(例 5)

【答案】[1] C1 [2] C1 [3] C2 [4] C3

【分析】本程序段主要体现了类 C1、C2、C3 和 C4 之间的层次关系，属于 C++ 面向对象程序设计框架。按照题目要求，类 C1 是整个层次结构的基类，类 C2 和 C3 都继承自 C1，因此 C2 和 C3 是 C1 的派生类。另一方面，类 C4 继承自 C2 和 C3，即 C4 同时具有两个基类，这种继承形式称为多继承。本例中整体的类层次结构如图 1.6 所示。

1.4 自 测 题

一、选择题

- 下列关于主函数的叙述中错误的是
 - 主函数是 C++ 程序执行的入口
 - 主函数的名称必须为 main
 - 每个 C++ 程序中都必须有且仅有一个主函数
 - 使用面向对象程序设计框架编写的 C++ 程序不需要主函数
- 下列选项中属于操作的是
 - 气球的颜色
 - 发动汽车
 - 桌子
 - 平面坐标系中的点 (1, 2)
- 下列关于 C++ 编译器和链接器的说法中正确的是
 - 编译器负责将源程序代码翻译成目标代码
 - 编译器输出的目标代码在功能上与源程序代码不完全等价
 - 链接器可以直接将源程序代码转变为可执行程序
 - 链接器输出的结果仍然不是可执行程序
- 下列关于面向对象基本概念的叙述中错误的是
 - 对象是系统中用来描述客观事物的一个实体
 - 类是对具有相同属性和相同操作的对象的定义
 - 继承是指使用新的类作为基础来覆盖已有的类
 - 多态是指同一种操作在不同对象之间可以表现出不同行为的现象

二、填空题

- C++ 中提供了“行”和“块”两种注释方法，行注释的内容以 [1] 开始直到本行末尾结束；块注释的内容从 [2] 开始到 [3] 结束。

2. C++ 标识符是以 [1] 开头的, 由 [2] 组成的字符串。
3. 在结构化程序设计框架中 [1] 是 C++ 程序的基本组成模块; 在面向对象程序设计框架中 [2] 是程序的基本组成模块。
4. 将下列程序补充完整, 使其在屏幕上输出字符串 "A C++ Program"。

```
#include <iostream>
int main()
{
    _____;
    return 0;
}
```

1.5 自测题答案

一、选择题

1. D 2. B 3. A 4. C

二、填空题

1. [1] // [2] /* [3] */
2. [1] 字母或下画线 [2] 字母、数字、下画线
3. [1] 函数 [2] 类
4. std::cout << "A C++ Program\n"

第2章 数据类型、运算符和表达式

2.1 学习目标与要求

1. 数据类型

- (1) 掌握定义基本数据类型及其派生类型的关键字。
- (2) 了解基本数据类型及其派生类型占用字节数和取值范围。
- (3) 理解不同的数据类型在程序中用来表示不同种类的数据。

2. 常量

- (1) 了解各种字面常量（逻辑常量、字符常量、整型常量、实型常量、枚举常量）的表示方法。
- (2) 能够正确使用 `const` 关键字定义符号常量。
- (3) 了解使用 `#define` 预处理命令也可以定义符号常量。

3. 变量

- (1) 能够正确定义各种类型的变量，掌握在变量定义时给其赋初值的方法。
- (2) 理解“变量对应于内存空间”的概念。
- (3) 理解使用变量时要遵循“先声明，后使用”的原则。
- (4) 了解全局变量和局部变量，生存期和作用域，变量存储类别的概念。

4. 运算符和表达式

- (1) 理解运算符和表达式的概念。
- (2) 掌握赋值运算符和复合赋值运算符的使用方法。
- (3) 熟练掌握算术运算符和算术表达式，关系运算符和关系表达式，逻辑运算符和逻辑表达式。
- (4) 理解前缀增量或减量运算符与后缀增量或减量运算符的区别。
- (5) 了解位运算符、条件运算符、逗号运算符、`sizeof` 运算符和圆括号运算符的作用。
- (6) 理解运算符优先级和结合性的概念，了解 C++ 运算符的优先级和结合性规则。

2.2 内容要点

一、数据类型

C++ 程序中的数据都具有一定的数据类型。使用类型可以确定数据占用的内存空间，数据

的取值范围，以及数据可以进行的运算种类。教程中对 C++ 数据类型的分类方式如图 2.1 所示。

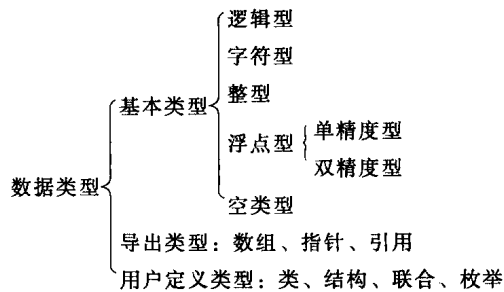


图 2.1 C++ 数据类型的分类

1. 基本类型

C++ 的基本数据类型描述了系统预定义的数据对象和可以对该类数据执行的操作。每种基本数据类型都使用一个关键字来表示，即：逻辑型 `bool`，字符型 `char`，整型 `int`，单精度浮点型 `float`，双精度浮点型 `double`，空类型 `void`。

2. 基本类型的变异

变异的类型仍属于 C++ 的基本数据类型，只是将基本类型的存储空间和取值范围放大或缩小，在基本类型关键字前面添加类型修饰符来表示。类型修饰符包括：`short`（短的）、`long`（长的）、`signed`（有符号的）、`unsigned`（无符号的）。

表 2.1 列出了在 VC6 中基本类型和其变异类型的存储空间及取值范围。

表 2.1 基本类型及变异类型的存储空间和取值范围

数 据 类 型	占用字节数	取 值 范 围	精 度
<code>bool</code>	1	true 或 false	
<code>char</code>	1	- 128 ~ 127	
<code>unsigned char</code>	1	0 ~ 255	
<code>int</code>	4	- $2^{31} \sim 2^{31} - 1$	
<code>unsigned int</code>	4	0 ~ $2^{32} - 1$	
<code>short int</code>	2	- $2^{15} \sim 2^{15} - 1$	
<code>unsigned short int</code>	2	0 ~ $2^{16} - 1$	
<code>long int</code>	4	- $2^{31} \sim 2^{31} - 1$	
<code>unsigned long int</code>	4	0 ~ $2^{32} - 1$	
<code>float</code>	4	- 3.4E+38 ~ 3.4E+38	至少 6 位数字
<code>double</code>	8	- 1.7E+308 ~ 1.7E+308	至少 15 位数字
<code>long double</code>	10	- 1.2E+4932 ~ 1.2E+4932	至少 19 位数字
<code>void</code>			

说明：

(1) 逻辑型 `bool` 是 C++ 中引入的基本类型（在 C 语言中没有此类型），它用关键字 `true` 表示“真”，用关键字 `false` 表示“假”。

(2) 表 2.1 中没有列出由 `signed`（有符号）修饰的类型，这是因为在默认情况下 `char` 和 `int` 就是有符号的，即：`char` 与 `signed char` 等价，`int` 与 `signed int` 等价。现将所有等价形式归纳如下：

<code>signed char</code>	$\Leftrightarrow$	<code>char</code>		
<code>signed int</code>	$\Leftrightarrow$	<code>int</code>		
<code>unsigned int</code>	$\Leftrightarrow$	<code>unsigned</code>		
<code>signed short int</code>	$\Leftrightarrow$	<code>short int</code>	$\Leftrightarrow$	<code>signed short</code> $\Leftrightarrow$ <code>short</code>
<code>unsigned short int</code>	$\Leftrightarrow$	<code>unsigned short</code>		
<code>signed long int</code>	$\Leftrightarrow$	<code>long int</code>	$\Leftrightarrow$	<code>signed long</code> $\Leftrightarrow$ <code>long</code>
<code>unsigned long int</code>	$\Leftrightarrow$	<code>unsigned long</code>		

二、常量

在程序运行过程中，其值不能被改变的量称为常量。常量有两种表示形式：

- (1) 字面常量：从其字面形式即可判断出所表示的常量值；
- (2) 符号常量：用标识符表示具体的常量值，即给字面常量取一个名字。

例如，“`pai` 等于 3.14159”，其中 `pai` 是符号常量，3.14159 是字面常量；“`LENGTH` 等于 50”，其中 `LENGTH` 是符号常量，50 是字面常量。

1. 字面常量

字面常量包括整型、实型、字符型、字符串型和逻辑型。

(1) 整型常量

整型常量有 3 种表示方式：

- ① 十进制整数。例如，123、-456、0。
- ② 八进制整数。以 0 开头的整数是八进制数。例如，0123 表示 8 进制数 $123_{(8)}$ ，等于十进制数 83。
- ③ 十六进制整数。以 0x 或 0X 开头的整数是十六进制数。例如，0x123 或 0X123 表示十六进制数 $123_{(16)}$ ，等于十进制数 291。

如果在整型常量后面加上字母 `U` 或 `u`，则表示无符号整数（`unsigned int`）。例如，123U 或 123u 是 `unsigned int` 型的。如果在整型常量后面加上一个字母 `L` 或 `l`，则表示长整数（`long int`）。例如，123L 或 123l 是 `long int` 型的。

(2) 实型常量

实型常量有 2 种表示方式：

- ① 十进制小数形式。由数字和小数点组成，且必须有小数点。例如，.123、123.、123.0、0.123、0.0 等。
- ② 指数形式。例如，123e4 或 123E4 都表示 123×10^4 。要注意 `e`（或 `E`）之前必须有数字，且 `e`（或 `E`）后面的指数必须为整数。如 `e4`、`1.2e3.4`、`.e4`、`e` 等都不是合法的指数形式。

如果没有任何说明, 一个实型常量表示 double 型数据; 要表示 float 型数据, 则必须在实型常量之后加上字母 F 或 f。例如, 1.23 是 double 型的, 而 1.23f 或 1.23F 是 float 型的。

(3) 字符常量

字符常量是由单引号括起来的字符。例如, 'a'、'A'、'? '、'\$'等都是字符常量。

具有控制功能或用作特殊标记的字符必须使用由反斜线引导的转义序列来表示, 例如, '\n' 表示换行符, '\t' 表示制表符, '\"' 表示单引号等, 教程第 19 页表 2.2 列出了 C++ 中常用的字符转义序列。

(4) 字符串常量

字符串常量是由一对双引号括起来的字符序列。例如。

```
"hello"
```

```
"I like the C++ programming language."
```

不要将字符常量与字符串常量混淆。'a' 是字符常量, "a" 是字符串常量, 二者不同。在 C++ 中, 字符串常量总是以 '\0' 结束。'a' 和 "a" 在内存中的存储形式分别如图 2.2 (a) 和 (b) 所示:

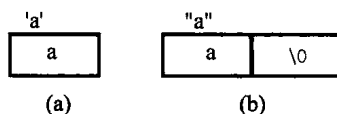


图 2.2 字符常量和字符串常量的存储形式

可以看出, 一个字符常量占用一个内存单元, 含有一个字符的字符串常量占用 2 个内存单元, 第 2 个内存单元存放 '\0' 结束符。

(5) 逻辑常量

C++ 语言使用两个关键字来表示逻辑常量, 即 true 代表逻辑“真”, false 代表逻辑“假”。

2. 符号常量

可以使用 const 关键字或 #define 预处理命令定义符号常量。

(1) 使用 const 定义符号常量

语法格式:

```
const <类型名> <符号常量名> = <初值>;
```

例如:

```
const double pai = 3.14159;
```

定义了符号常量, 它用来表示字面常量 3.14159。

注意: 声明符号常量时必须为其赋初值, 且该值在程序中不能被改变。

(2) 使用 #define 定义符号常量

语法格式:

```
#define <符号常量名> <字符序列>
```

例如:

```
#define pai 3.14159
```

定义了符号常量, 它用来表示字面常量 3.14159。

C++ 允许使用 #define 定义符号常量主要是为了兼容 C 语言, 使 C 程序能在 C++ 编译器中顺利通过。在 C++ 程序中, 定义符号常量一般都使用 const, 而不用 #define。

(3) 枚举常量

枚举常量实际上是一组含义相关的整型符号常量, 它可以通过建立枚举类型来定义。声明枚