

# C++ 习题与解答

李代平 张伯泉 编著



冶金工业出版社

# C++习题与解答

李代平 张伯泉 编著

北 京

冶金工业出版社

2003

## 内 容 简 介

本书配合《C++程序设计教程》的内容，介绍了数据类型与表达式、控制结构、函数、数组、指针、结构体与共用体、类和对象、继承和派生类、多态、类属机制、C++的输入/输出流库和异常处理等内容，并就各章节知识点给出了综合练习，还给出了相关练习的参考答案。

本书可以作为大专院校相关专业学生的参考用书，特别适合计算机本科和专科的学生、报考计算机专业硕士研究生的考生、参加国家高等教育自学考试的考生、参加计算机等级考试的考生以及计算机专业的高级人员参考。

### 图书在版编目 (CIP) 数据

C++习题与解答 / 李代平等编著. —北京: 冶金工业出版社, 2003.6

ISBN 7-5024-3282-5

I. C... II. 李... III. C语言—解题  
IV. TP391-44

中国版本图书馆 CIP 数据核字 (2003) 第 035008 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

中山市新华印刷厂有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2003 年 7 月第 1 版, 2003 年 7 月第 1 次印刷

787mm × 1092mm 1/16; 17.5 印张; 414 千字; 272 页; 1-5000 册

25.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

# 前 言

## 一、关于本书

C++是一门含C语言子集的高效程序设计语言，它既可以进行过程化程序设计，也可以用于面向对象的程序设计。本书是《C++程序设计教程》的配套用书，每章的内容都包括基础知识、单元练习和参考答案这三大部分，是作者根据多年的教学经验整理而成的。

## 二、本书结构

本书共由13章组成：

第1章：绪论。主要内容包括：面向对象的主要概念、C与C++的关系。

第2章：数据类型与表达式。主要内容包括：数据类型、常量、变量、存储类型说明、运算符、控制符以及数据的输入/输出。

第3章：控制结构。主要内容包括：C++语句、分支结构和循环结构。

第4章：函数。主要内容包括：C++语言程序组件、函数的定义与声明以及函数的调用。

第5章：数组。主要内容包括：数组的定义、数组的初始化、数组的引用以及数组与函数、常用串处理函数。

第6章：指针。主要内容包括：指针的概念、指针变量的定义和引用、指针运算、指针作函数参数、指针与数组、指针与函数、指向指针的指针、main函数中的参数。

第7章：结构体与共用体。主要内容包括：结构体类型的定义、结构体类型变量或数组的定义、结构体变量的初始化、结构体类型变量的引用、位段、共用体。

第8章：类和对象。主要内容包括：类、对象、this指针、友元函数。

第9章：继承和派生类。主要内容包括：基类和派生类，派生与构造函数、析构函数，虚基类。

第10章：多态。主要内容包括：重载运算符、函数重载、虚函数、抽象类。

第11章：类属机制。主要内容包括：函数模板、类模板。

第12章：C++的I/O流库。主要内容包括：I/O标准流类、键盘输入、屏幕输出、格式化输入和输出、字符串流。

第13章：异常处理。主要内容包括：异常的概念、基本原理、异常的处理方法以及多路捕获。

## 三、本书特点

本书每章包括的三个题型（选择题、填空题、编程题）是与《C++程序设计教程》相对应的。选择题、填空题考察读者的基本知识，对选择题给出了必要的分析，力求对每个知识点都讲解清楚，使读者对该知识点的来龙去脉有较深刻的理解；填空题进一步巩固基本知识点；编程题考察读者的综合知识的运用能力，对每道编程题都给出了参考答案，这些参考答案都能够在Visual C++编译器上编译成功，具有参考性和启发性，读者也可参考答案另行编程。通过本书的学习，读者能够较全面的掌握C++语言的相关知识，并能较快地提高自己的编程水平。

#### 四、适用对象

本书可以作为大专院校相关专业学生的参考用书,特别适合于计算机本科和专科的学生、报考计算机专业硕士研究生的考生、参加国家高等教育自学考试的考生、参加计算机等级考试的考生以及计算机专业的高级人员参考。

由于 C++语言知识面广,内容深,加以时间仓促,再加上作者水平有限,书中的不足之处在所难免,恳请读者批评指正。

读者如果有好的意见或建议,可以到相关网站进行探讨。**网址: <http://www.cnbook.net>。**

编者

2003年5月

# 目 录

<b>第1章 绪论</b> .....	<b>1</b>
1.1 基础知识 .....	1
1.1.1 面向对象的主要概念 .....	1
1.1.2 C与C++的关系 .....	3
1.2 单元练习 .....	3
1.2.1 选择题 .....	3
1.2.2 填空题 .....	5
1.2.3 编程题 .....	5
1.3 参考答案 .....	6
<b>第2章 数据类型与表达式</b> .....	<b>8</b>
2.1 基础知识 .....	8
2.1.1 数据类型 .....	8
2.1.2 常量 .....	8
2.1.3 变量 .....	8
2.1.4 存储类型说明 .....	9
2.1.5 运算符及其优先级 .....	9
2.1.6 控制符 .....	10
2.1.7 数据的输入/输出 .....	11
2.2 单元练习 .....	11
2.2.1 选择题 .....	11
2.2.2 填空题 .....	13
2.2.3 编程题 .....	15
2.3 参考答案 .....	16
<b>第3章 控制结构</b> .....	<b>19</b>
3.1 基础知识 .....	19
3.1.1 C++语句 .....	19
3.1.2 分支结构 .....	19
3.1.3 循环结构 .....	21
3.2 单元练习 .....	22
3.2.1 选择题 .....	22
3.2.2 填空题 .....	30
3.2.3 编程题 .....	33
3.3 参考答案 .....	33
<b>第4章 函数</b> .....	<b>40</b>
4.1 基础知识 .....	40
4.1.1 C++语言程序组件 .....	40

4.1.2 函数的定义与声明 .....	41
4.1.3 函数的调用 .....	42
4.2 单元练习 .....	42
4.2.1 选择题 .....	42
4.2.2 填空题 .....	50
4.2.3 编程题 .....	53
4.3 参考答案 .....	54
<b>第5章 数组 .....</b>	<b>67</b>
5.1 基础知识 .....	67
5.1.1 数组的定义 .....	67
5.1.2 数组的初始化 .....	68
5.1.3 数组的引用 .....	69
5.1.4 数组与函数 .....	70
5.1.5 常用串处理函数 .....	70
5.2 单元练习 .....	70
5.2.1 选择题 .....	70
5.2.2 填空题 .....	78
5.2.3 编程题 .....	81
5.3 参考答案 .....	82
<b>第6章 指针 .....</b>	<b>92</b>
6.1 基础知识 .....	92
6.1.1 指针的概念 .....	92
6.1.2 指针变量的定义和引用 .....	92
6.1.3 指针运算 .....	93
6.1.4 指针作函数参数 .....	94
6.1.5 指针与数组 .....	94
6.1.6 指针与函数 .....	94
6.1.7 指向指针的指针 .....	94
6.1.8 main 函数中的参数 .....	95
6.2 单元练习 .....	95
6.2.1 选择题 .....	95
6.2.2 填空题 .....	110
6.2.3 编程题 .....	112
6.3 参考答案 .....	112
<b>第7章 结构体与共用体 .....</b>	<b>126</b>
7.1 基础知识 .....	126
7.1.1 结构体类型的定义 .....	126
7.1.2 结构体类型变量或数组的定义 .....	126
7.1.3 结构体变量的初始化 .....	127

7.1.4 结构体类型变量的引用 .....	127
7.1.5 结构体类型数据的指针 .....	127
7.1.6 位段 .....	128
7.1.7 共用体 .....	128
7.2 单元练习 .....	128
7.2.1 选择题 .....	128
7.2.2 填空题 .....	137
7.2.3 编程题 .....	140
7.3 参考答案 .....	140
<b>第 8 章 类和对象 .....</b>	<b>148</b>
8.1 基础知识 .....	148
8.1.1 概述 .....	148
8.1.2 类 .....	148
8.1.3 对象 .....	149
8.1.4 对象的初始化 .....	149
8.1.5 成员函数的重载 .....	150
8.1.6 this 指针 .....	150
8.1.7 友元函数 .....	150
8.1.8 静态成员 .....	150
8.2 单元练习 .....	150
8.2.1 选择题 .....	150
8.2.2 填空题 .....	163
8.2.3 编程题 .....	164
8.3 参考答案 .....	165
<b>第 9 章 继承和派生类 .....</b>	<b>179</b>
9.1 基础知识 .....	179
9.1.1 基类和派生类 .....	179
9.1.2 派生与构造函数、析构函数 .....	180
9.1.3 虚基类 .....	180
9.2 单元练习 .....	181
9.2.1 选择题 .....	181
9.2.2 填空题 .....	190
9.2.3 编程题 .....	193
9.3 参考答案 .....	194
<b>第 10 章 多态 .....</b>	<b>205</b>
10.1 基础知识 .....	205
10.1.1 重载运算符 .....	205
10.1.2 函数重载 .....	206
10.1.3 虚函数 .....	206



10.1.4 抽象类.....	207
10.2 单元练习.....	208
10.2.1 选择题.....	208
10.2.2 填空题.....	218
10.2.3 编程题.....	220
10.3 参考答案.....	221
<b>第 11 章 类属机制.....</b>	<b>231</b>
11.1 基础知识.....	231
11.1.1 函数模板.....	231
11.1.2 类模板.....	232
11.2 单元练习.....	233
11.2.1 选择题.....	233
11.2.2 填空题.....	236
11.2.3 编程题.....	236
11.3 参考答案.....	236
<b>第 12 章 C++的 I/O 流库.....</b>	<b>243</b>
12.1 基础知识.....	244
12.1.1 I/O 标准流类.....	244
12.1.2 键盘输入.....	244
12.1.3 屏幕输出.....	244
12.1.4 插入符和提取符的重载.....	245
12.1.5 磁盘文件的输入和输出.....	245
12.1.6 格式化输入和输出.....	246
12.1.7 字符串流.....	246
12.2 单元练习.....	246
12.2.1 选择题.....	246
12.2.2 填空题.....	252
12.2.3 编程题.....	255
12.3 参考答案.....	255
<b>第 13 章 异常处理.....</b>	<b>264</b>
13.1 基础知识.....	264
13.1.1 异常的概念.....	264
13.1.2 基本原理.....	264
13.1.3 异常的处理方法.....	265
13.1.4 多路捕获.....	266
13.2 单元练习.....	266
13.2.1 选择题.....	266
13.2.2 填空题.....	268
13.3 参考答案.....	271

# 第1章 绪 论

计算机程序设计语言经历了从低级语言到高级语言的发展过程。C++是一种面向对象的程序设计语言，同时也包含C语言子集，也就是说用它既可以进行结构化程序设计，又可以进行面向对象的程序设计，是一种既面向对象又面向过程的混合设计语言。为了对面向对象的程序设计有一个基本的了解，本章介绍了面向对象的基本概念、C与C++的关系等知识，希望读者对C++有一个初步的了解，为进一步的学习奠定基础。

## 1.1 基础知识

### 1.1.1 面向对象的主要概念

面向对象的系统主要包含：对象、类、封装、继承、多态性和抽象等要素和概念。这些要素和概念反映了面向对象的传统观念。

#### 1. 对象

在面向对象的领域中，应从两个角度来理解对象这个名词，一是它的概念，二是它的实现。从概念上讲，它是系统中的一个实体，比如一个汽车修理厂中的汽车、修理员等都是对象。从一般意义上讲，对象是现实世界中一个实际存在的事物，它可以是有形的（比如一辆汽车），也可以是无形的（比如一项计划）。从实现形式上讲，对象是属性和服务的封装体，由一组属性和对这组属性进行操作的一组服务构成。它是构成世界的一个独立单位，具有自己的静态特征和动态特征。静态特征即属性可以用某种数据来描述的特征，动态特征即对象所表现的行为或对象所具有的功能或方法。

属性和服务，是构成对象的两个主要因素。属性是用来描述对象静态特征的一个数据项。服务是用来描述对象动态特征（行为）的一个操作序列。

一个对象可以有多个属性和多个服务。一个对象的属性和服务被结合成一个整体，对象的属性值只能由这个对象的服务存取。

对象实现了信息隐藏，对象与外部通过接口联系，服务的具体实现对外部来说是不可见的。封装的目的就是阻止外界的非法访问。对象之间的联系是通过这些接口来启动的。对象的属性和服务是紧密相连的。属性的改变需服务来进行。

#### 2. 类

类是具有相同属性和服务的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和服务两个主要部分。例如：马、树木、石头等等都是一些抽象概念，它们是一些具有共同特征的事物的集合，被称作类。类的概念使读者能对属于该类的全部个体事物进行统一的描述。例如，“树具有树根、树干、树枝和树叶，它能进行光合作用”，这个描述适合所有的树，从而不必对每棵具体的树都进行一次这样的描述。

类与对象的关系如同一个模具与用这个模具铸造出来的铸件之间的关系。类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。所以，一个对象又称作类的一个实例（instance）。所谓“实体”、“实例”意味着：在程序中，每个对象需要有自己

的存储空间，以保存它们自己的属性值。通常说同类对象具有相同的属性与服务，是指它们的定义形式相同，而不是说每个对象的属性值都相同。在程序中类是不占有空间的，用类定义的对象才分配空间。这点需要读者注意，也是抽象与现实之间的一个联系点。

### 3. 封装

封装是面向对象方法的一个重要原则。它有两个涵义：第一个涵义是把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位（即对象）。第二个涵义也称作“信息隐蔽”，即尽可能隐蔽对象的内部细节，对外形成一个边界（或者说形成一道屏障）、只保留有限的对外接口使之与外部发生联系，这主要是指对象的外部不能直接地存取对象的属性，只能通过几个允许外部使用的服务与对象发生联系。

例如：用“售报亭”对象描述现实中的一个售报亭。它的属性是亭内的各种报刊（其名称、定价）和钱箱（总金额），它有两个服务——报刊零售和货款清点。当用户去买一份报纸时，不可能直接去拿报纸，只有把这个买报纸的这一意图和所需金额交给卖方，卖方通过零售这一服务才能实现用户买报纸这一愿望。

### 4. 继承

特殊类的对象拥有其一般类的全部属性与服务，称作特殊类对一般类的继承。

一个特殊类既有自己新定义的属性和服务，又有从它的一般类中继承下来的属性与服务。继承来的属性服务，尽管是隐式的（不用书写出来），但是无论在概念上还是在实际效果上，都确实实地是这个类的属性和服务。当这个特殊类又被它更下层的特殊类继承时，它继承来的和自己定义的属性和服务又都一起被下层的类继承下去。也就是说，继承关系是传递的。

继承具有重要的实际意义，它简化了人们对事物的认识和描述。比如认识了轮船的特征之后，在考虑客轮时，只要知道客轮也是一种轮船这个事实，那就认为它理所当然的具有轮船的全部一般特征，只需要把精力用于发现和描述客轮独有的那些特征上。在软件开发过程中，在定义特殊类时，不需把它的一般类已经定义过的属性和服务重复地书写一遍，只需要声明它是某个类的特殊类，并定义它自己的特殊属性和服务，无疑这将明显地减轻开发工作的强度。

### 5. 多态性

对象的多态性是指在一般类中定义的属性或服务被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或服务名在一般类及其各个特殊类中具有不同的语义。

例如，在一般类“几何图形”中定义了一个服务“绘图”，但并不确定执行时到底画了一个什么图形。特殊类“椭圆”和“多边形”都继承了几何图形类的绘图服务，但其功能却不同：一个是画出一个椭圆，一个是画出一个多边形。进而，在多边形类更下层的一般类“矩形”中绘图服务又可以采用一个比画一般的多边形更高效的算法来画一个矩形。这样，当系统的其余部分请求画出任何一种几何图形时，消息中给出的服务名同样都是“绘图”（因而消息的书写方式可以统一），而椭圆、多边形、矩形等类的对象接收到这个消息时却各自执行不同的绘图算法。

### 6. 抽象

从许多事物中舍弃个别的、非本质性的特征，抽取共同的、本质性的特征，就叫做抽象（abstraction）。抽象是形成概念的必要手段。

抽象, 只需要分析研究其中与系统目标有关的事物及其本质性特征, 对于那些与系统目标无关的特征和许多具体的细节, 即使有所了解, 也应该舍弃。通过舍弃个体事物在细节的差异, 抽取其共同特征而得到一批事物的抽象概念。抽象是面向对象方法中使用最为广泛的原则。

### 1.1.2 C 与 C++的关系

C 语言是 C++的一个子集, C++包含了 C 的全部。

#### 1. C++保持与 C 的兼容

兼容性的表现使许多 C 语言代码不需修改就可以为 C++所用。以前用 C 语言编写的许多库函数和应用软件都可以用于 C++。

#### 2. C++对 C 作了很多改进

C++保持了 C 的简洁与接近汇编的特点, 同时又有重要的改进。

(1) 增加了一些运算符, 如: new, delete 等。

(2) 改进了类型系统, 增加了安全性。C++规定数据类型转换采用强制转换。规定函数的说明必须采用原型。对于缺省函数作了限制。增加了编译系统检查类型的能力。

(3) 引进了引用概念, 将引用作为函数参数带来很大方便。

(4) 允许函数重载, 允许设置缺省参数, 这些措施提高了程序设计的灵活性。引入内联函数的概念, 提高了程序设计的有效性。

(5) 对于变量更加灵活。在 C 语言中仅允许变量在函数体内或分段程序内。而且对变量必须先说明后使用, 不能交叉使用, C++打破了这一规定, 可以根据要求对变量进行说明。

#### 3. C++与 C 的差别

C++与 C 语言的差别在于 C 是面向过程的, C++是面向对象的。因此, C++是在对 C 语言改造的基础上又添加了面向对象的内容。

## 1.2 单元练习

### 1.2.1 选择题

1. C++是( )。

- A. 面向对象的程序设计语言
- B. 结构化程序设计语言
- C. 既是面向对象的程序设计语言又是结构化程序设计语言
- D. 非结构化程序设计语言

2. 下列关于对象的描述不正确的是( )。

- A. 对象只能描述有形的东西
- B. 对象具有属性和服务两个主要因素
- C. 现实世界中的一切事物都可以看作对象, 但只有在具体的系统中加以论述才有实际的意义
- D. 对象的属性是对象的静态特征, 服务是动态特征, 并且对象的属性只能由本对象的服务来操作



C. 语句的构成不一定需要分号

D. 语句的书写对程序的运行不起作用, 因此可以随便写都可以

### 1.2.2 填空题

1. 构成对象的两个主要因素是\_\_\_\_\_和\_\_\_\_\_, 其中\_\_\_\_\_属于动态属性, \_\_\_\_\_属于静态属性, \_\_\_\_\_只能由\_\_\_\_\_来操作。
2. 类是\_\_\_\_\_的集合, 分类的依据是\_\_\_\_\_。
3. 程序的三种基本的控制结构是: \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
4. 算法具有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_的性质, 所以在编程时要保证这些性质的实现。
5. 这是一个从键盘输入三个数, 找出最大数的程序。请在空白处填写所缺内容。

```

main ( )
{
    int a,b,c,x;
    cin>>a;
    cout<<"a="<<a<<endl;
    cin>>b;
    cout<<"b="<<b<<endl;
    cin>>c;
    cout<<"c="<<c<<endl;

    if_____
        cout<<"the three numbers are equal,they are:<<a<<endl";
    else
    {
        if ( a>b)
            x=a;
        else
            if_____
                x=b;
            else
                _____
        cout<<"the greatest number is :"<<x;
    }
}

```

### 1.2.3 编程题

1. 编写一个 C++ 程序, 打印出如下的图形。

```

*
*
*
*****
*
*
*

```

2. 输入三个整数, 按从小到大的顺序打印出来。
3. 编一程序实现在屏幕上输出: Have a good time。

### 1.3 参考答案

#### 一、选择题

1. C C++是在C语言的基础上发展起来的，它保留了C语言面向过程的特点，又添加了面对对象的特点。
2. A 正确理解对象的含义，抽象是寻找对象的基本方法。
3. C 在宇宙系统中把太阳系看作对象比较合理，地球、月亮、行星包含在太阳系内。
4. C 细菌不属于动物类。
5. D 注意C++中合法标识符的规定。
6. C “A”与‘A’占用内存数不同：前者占用两个字节，后者占用一个字节。
7. D main()函数是可以有参数的，它的参数有其独特的含义。
8. C 正确理解封装的含义。
9. C 正确理解继承的含义。
10. B 注意复合语句与单语句的区别；一条语句必以分号结束；语句书定应规整，易于理解。

#### 二、填空题

1. 属性      服务      服务      属性      属性      服务
2. 具有相同属性和服务的一组对象      抽象
3. 顺序结构              循环结构              选择结构
4. 有穷性      一个初始操作      一个动作只有一个惟一后继      一个终止回答
5. #include <iostream.h>或#include "iostream.h"（因为程序中的输入函数cin、输出函数cout是在头文件iostream.h中说明的，所以要有iostream.h头文件。以后在编程时候会常常用到一些头文件中所包含的函数，这时要把包含该函数的头文件加在程序文件的首部即可，至于所用的函数具体包含在哪个头文件中可以查阅相关资料）  
a==b&&a==c 或 a==b&&b=c  
或 a==c&&b==c（“=”与“==”不同，前者是赋值函数，后者是关系运算符，&&是逻辑与运算符，其功能可以查看第2章）      b>c      x=c

#### 三、编程题

1.

```
#include <iostream.h>
main ( )
{
    cout<<"      *"<<endl;           // "<<endl" 用于输出换行
    cout<<"      *"<<endl;
    cout<<"      *"<<endl;           //本程序可用循环控制结构处理，可使程序简洁，请试写
    cout<<"      *****"<<endl;
    cout<<"      *"<<endl;
    cout<<"      *"<<endl;
    cout<<"      *"
```

2.

```
#include <iostream.h>
main ( )
```

```
{ int a,b,c,x;
  cout<>a>>b>>c;
  if ( a>b)
  { x=a;
    a=b;
    b=x;}
  if ( a>c)
  { x=c;
    c=a;
    a=x;}
  if ( b>c)
  { x=c;
    c=b;
    b=x;}
  cout<< a<< b<< c;}
```

// “ a” 这种写法是为了数据与数据间有空隙

3.

```
#include <iostream.h>
void main ( )
{
  cout<<"    Have a good time";
}
```



## 第 2 章 数据类型与表达式

计算机的基本功能是进行数据处理。在 C++ 语言中, 数据处理的基本对象是常量和变量。运算是对各种形式的数据进行处理, C++ 语言的运算功能十分丰富, 它既具有算术运算、逻辑运算和关系运算等一般的常用运算功能, 又可以对数据进行位移操作处理和某些特殊运算。本章介绍数据类型和运算的种类及其特点。

### 2.1 基础知识

#### 2.1.1 数据类型

C++ 提供的数据类型有: 基本类型、构造类型和指针类型。

其中基本类型包括整型、实型(浮点型)、字符型、逻辑型。

构造类型包括数组类型、结构类型、联合型类型、枚举型类型, 如图 2-1 所示。

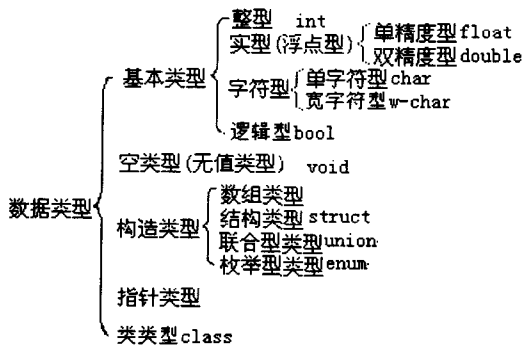


图 2-1

#### 2.1.2 常量

常量是指程序在运行时其值不能改变的量, 常量在程序中一般以自身存在形式体现其值, 也可通过宏定义命令来定义, 即用一个标识符来代替某个常量。常量具有类型属性, 类型决定了各种常量在内存中所占存储空间的大小。C++ 语言规定的常量类型如图 2-2 所示。

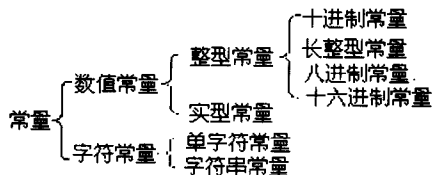


图 2-2

#### 2.1.3 变量

变量是指程序在运行时其值可改变的量。变量在内存中占有固定的存储单元, 并有确定的变量名。它用来存储各种类型的数据, 以及指向存储器内部单元的指针。