

清华大学计算机基础教育课程系列教材

C++语言程序设计习题与实验指导

版社

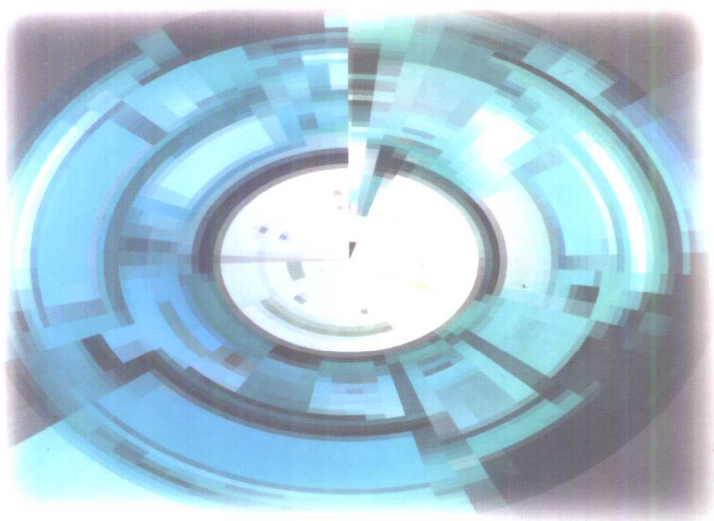


清华大学计算机基础教育课程系列教材

# C++语言程序设计

## 习题与实验指导

郑莉 傅仕星 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



清华大学计算机基础教育课程系列教材

# C++ 语言程序设计

## 习题与实验指导

郑莉 傅仕星 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

**(京)新登字 158 号**

### 内 容 简 介

本书是“清华大学计算机基础教育课程系列教材”中的《C++ 语言程序设计》一书的配套教材。书中内容分为两部分:第 1 部分“习题解答”给出了《C++ 语言程序设计》各章习题的参考答案以及少量的补充习题和答案,共 186 道。第 2 部分“实验指导”包括 13 个精心设计的实验,分别与《C++ 语言程序设计》各章的内容相配合。每个实验均包括实验目的、实验任务、实验步骤等内容。书后的三个附录分别是“Visual C++ 开发环境简介”、“ASCII 码表”和“常用系统函数”。随书的软盘中附有本书中所有实验的参考程序和《C++ 语言程序设计》一书中的例题程序的源代码。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

**书 名:** C++ 语言程序设计习题与实验指导

**作 者:** 郑 莉 傅仕星 编著

**出 版 者:** 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

**责任编辑:** 石磊(E-mail:shl@tup.tsinghua.edu.cn)

**印 刷 者:** 北京市人民文学印刷厂

**发 行 者:** 新华书店总店北京发行所

**开 本:** 787×1092 1/16 **印 张:** 13.75 **字 数:** 312 千字

**版 次:** 2000 年 7 月第 1 版 2000 年 12 月第 2 次印刷

**书 号:** ISBN 7-302-03926-7/TP·2294

**印 数:** 8001~14000

**定 价:** 25.00 元(含盘)

# 序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广：该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革，该系列教材中多数都配有习题集和实验指导、多媒体电子教案，有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践，凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善，不断更新，为我国高校计算机基础教育做出新的贡献。



1999 年 12 月

---

注：周远清，现任教育部副部长，原清华大学副校长、计算机专业教授。

# 前言

计算机程序设计是一门实践性很强的课程,因此仅仅通过阅读教科书或听课是不可能掌握的。学习程序设计最重要的环节就是实践。

凡是学习程序设计的人,往往有这样的感觉:看书或听课时,对老师讲的和书上写的内容基本上能够理解,但是当需要自己编程时却又无从下手。相信每一个讲授程序设计课程的教师都有过这样的经历:有些问题,尽管我们在课上再三强调,反复举例,学生还是不能够完全理解,上机时更是错误百出。应该说,这是学习过程中的必然现象。要想能够把书本上的知识变为自己所具有的能力,所需要的是实践、实践、再实践。在实践环节中,起主导作用的是学习者自己,旁人是无法代劳的,也不能期望有什么一蹴而就的捷径。但是由于学生在实践过程中不能随时随地得到指导,因此花费时间较多,总感觉程序设计课作业负担太重,有的学生甚至因为花四、五个小时调不通一个简单的程序而失去学习兴趣。像C++这样面向对象的程序设计语言学习起来尤其如此。笔者编写本书作为与《C++语言程序设计》配套的辅导与实验教材,目的就在于为读者从看懂教科书过渡到具有基本的编程能力助一臂之力,使读者在实践的过程中少些曲折和彷徨,多些成功的乐趣。本书的内容已经在清华大学非计算机专业的“计算机软件技术基础”课中试用,取得了良好的教学效果。

书中内容分为两部分:第1部分“习题解答”给出了《C++语言程序设计》各章习题的参考答案以及少量补充习题和答案,共186道。每个题目可能有多种解法,我们仅给出一种参考解法。大部分题目是编程题,我们在解答中给出了其主要程序段的源程序清单,其中有些不是完整的程序,如果需要运行,只需将它们插入调试程序即可。第2部分“实验指导”包括13个精心设计的实验,分别与《C++语言程序设计》各章内容配合,使读者在实践中达到对书中内容的深入理解和熟练掌握。每一个实验都包括“实验目的”、“实验任务”和“实验步骤”。此外,书后还有三个附录。附录A“Visual C++开发环境简介”简单明了地介绍了Visual C++集成开发环境及其使用方法,其中包括使用Visual C++的应用程序向导创建一个项目的详细步骤。附录B是“ASCII码表”。附录C“常用系统函数”列出了C++中的常用库函数及其说明,可作为简明函数手册使用。

本书中的习题解答和实验内容不仅可以指导读者上机练习,也可以由教师选做例题在课堂上演示,使教学内容更加丰富。如果读者没有足够的时间一一做完全部习题和实验,将题解作为例题阅读也不失为一种好的选择。不过在此仍要建议,读者在学习C++时尽量自己完成习题和实验程序,经过独立思考之后再参考本书中的答案,以开阔思路,不要一味依赖解答。只有经过独立思考和实践,才能真正提高编程能力。如果在编写习题和实验程序时遇到了难以解决的困难,本书中的答案可以提供参考。

本书中的全部程序都在Windows环境下Visual C++开发环境中测试通过。鉴于实

验指导部分的答案程序多数都比较长,而且在很多地方引用了《C++ 语言程序设计》一书中的例题,为了方便读者,我们将实验参考程序和《C++ 语言程序设计》一书中的例题程序都附在随书的软盘中。

参加本书编写和文字工作的还有董渊、田荣牌、孟鸿俐,在此深表感谢。

感谢读者选择使用本书,欢迎您对本书内容提出批评和修改建议,我们将不胜感激,并在再版时予以考虑。作者的联系地址如下:

电子邮件地址: zhli@mailx.cic.tsinghua.edu.cn

通信地址: 北京 清华大学计算机与信息管理中心(主楼 217) 郑莉 收

邮政编码: 100084

作 者

2000 年 2 月于清华大学

# 目 录

<b>第 1 部分 习题解答</b> .....	1
第 1 章 概述.....	3
第 2 章 C++简单程序设计 .....	6
第 3 章 函数 .....	22
第 4 章 类与对象 .....	31
第 5 章 C++程序的结构.....	39
第 6 章 数组、指针与字符串.....	46
第 7 章 继承与派生 .....	60
第 8 章 多态性 .....	69
第 9 章 群体类 .....	81
第 10 章 群体数据的组织.....	99
第 11 章 流类库与输入/输出.....	109
第 12 章 异常处理 .....	115
第 13 章 MFC 库与 Windows 程序开发概述 .....	120
<b>第 2 部分 实验指导</b> .....	121
实验 1 Visual C++ 6.0 开发环境应用入门(2 学时) .....	123
实验 2 C++简单程序设计(4 学时).....	126
实验 3 函数的应用(2 学时) .....	131
实验 4 类与对象(4 学时) .....	134
实验 5 C++程序的结构(2 学时).....	137
实验 6 数组、指针与字符串(4 学时).....	139
实验 7 继承与派生(4 学时) .....	141
实验 8 多态性(2 学时) .....	143
实验 9 群体类(4 学时) .....	144
实验 10 群体数据的组织(2 学时).....	146
实验 11 流类库与输入/输出(2 学时) .....	147
实验 12 异常处理(2 学时).....	150
实验 13 Windows 应用程序框架(2 学时) .....	151
附录 A Visual C++ 开发环境简介 .....	152
附录 B ASCII 码表 .....	166
附录 C 常用系统函数 .....	168



# 第 1 部分

# 习 题 解 答



# 第 1 章

## 概述

1-1 简述计算机程序设计语言的发展历程。

解：迄今为止,计算机程序设计语言的发展经历了机器语言、汇编语言、高级语言等阶段。C++语言是一种面向对象的编程语言,也属于高级语言。

1-2 面向对象的编程语言有哪些特点?

解：面向对象的编程语言与以往各种编程语言有根本的不同,它设计的出发点就是为了能更直接地描述客观世界中存在的事物以及它们之间的关系。面向对象的编程语言将客观事物看作具有属性和行为的对象,通过抽象找出同一类对象的共同属性(静态特征)和行为(动态特征),形成类。通过类的继承与多态可以很方便地实现代码重用,大大缩短了软件开发周期,并使软件风格统一。因此,面向对象的编程语言使程序能够比较直接地反映问题域的本来面目,使软件开发人员能够利用人类认识事物所采用的一般思维方法进行软件开发。C++语言是目前应用最广的面向对象的编程语言。

1-3 什么是结构化程序设计方法?这种方法有哪些优点和缺点?

解：结构化程序设计方法的思路是:自顶向下、逐步求精。其程序结构是按功能划分为若干个基本模块;各模块之间的关系尽可能简单,在功能上相对独立;每一模块内部均是由顺序、选择和循环三种基本结构组成。其模块化实现的具体方法是使用子程序。结构化程序设计由于采用了模块分解与功能抽象和自顶向下、分而治之的方法,从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务,便于开发和维护。

虽然结构化程序设计方法具有很多的优点,但它仍是一种面向过程的程序设计方法,它把数据和处理数据的过程分离为相互独立的实体。当数据结构改变时,所有相关的处理过程都要进行相应的修改,每一种相对于老问题的新方法都要带来额外的开销,程序的可重用性差。

由于图形用户界面的应用,程序运行由顺序运行演变为事件驱动,使得软件使用起来越来越方便,但开发起来却越来越困难,对这种软件的功能很难用过程来描述和实现,使用面向过程的方法来开发和维护此类软件都将非常困难。

1-4 什么是对象?什么是面向对象方法?这种方法有哪些特点?

解：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,也可以

是无形的。对象是构成世界的一个独立单位,它具有自己的静态特征和动态特征。面向对象方法中的对象,是系统中用来描述客观事物的一个实体,它是用来构成系统的一个基本单位,由一组属性和一组行为构成。

面向对象的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讯。这样,程序模块间的关系更为简单,程序模块的独立性、数据的安全性就有了良好的保障。通过实现继承与多态性,还可以大大提高程序的可重用性,使得软件的开发和维护都更为方便。

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。

#### 1-5 什么叫做封装?

**解:** 封装是面向对象方法的一个重要原则,就是把对象的属性和服务结合成一个独立的系统单位,并尽可能隐蔽对象的内部细节。

#### 1-6 面向对象的软件工程包括哪些主要内容?

**解:** 面向对象的软件工程是面向对象方法在软件工程领域的全面应用,它包括面向对象的分析(OOA)、面向对象的设计(OOD)、面向对象的编程(OOP)、面向对象的测试(OOT)和面向对象的软件维护(OOSM)等主要内容。

#### 1-7 计算机内部的信息可分为几类?

**解:** 计算机内部的信息可以分成控制信息和数据信息两大类;控制信息可分为指令和控制字两类;数据信息可分为数值信息和非数值信息两类。

#### 1-8 什么叫二进制? 使用二进制有何优点和缺点?

**解:** 二进制是基数为 2,每位的权是以 2 为底的幂的进制,遵循逢二进一原则,基本符号为 0 和 1。采用二进制码表示信息,有如下几个优点:(1)易于物理实现;(2)二进制数运算简单;(3)机器可靠性高;(4)通用性强。其缺点是它表示数的容量较小,表示同一个数,二进制较其他进制需要更多的位数。

#### 1-9 请将以下十进制数值转换为二进制和十六进制补码:

- (1)2            (2)9            (3)93  
(4)-32        (5)65535        (6)-1

**解:**

- (1)    (2)<sub>16</sub> = (10)<sub>2</sub> = (2)<sub>16</sub>  
(2)    (9)<sub>16</sub> = (1001)<sub>2</sub> = (9)<sub>16</sub>

$$(3) \quad (93)_{10} = (1011101)_2 = (5D)_{16}$$

$$(4) \quad (-32)_{10} = (11100000)_2 = (E0)_{16}$$

$$(5) \quad (65535)_{10} = (11111111 \ 11111111)_2 = (FFFF)_{16}$$

$$(6) \quad (-1)_{10} = (11111111 \ 11111111)_2 = (FFFF)_{16}$$

**1-10** 请将以下数值转换为十进制：

$$(1) (1010)_2 \quad (2) (10001111)_2 \quad (3) (01011111 \ 11000011)_2$$

$$(4) (7F)_{16} \quad (5) (2D3E)_{16} \quad (6) (F10E)_{16}$$

**解：**

$$(1) \quad (1010)_2 = (10)_{10}$$

$$(2) \quad (10001111)_2 = (143)_{10}$$

$$(3) \quad (01011111 \ 11000011)_2 = (24515)_{10}$$

$$(4) \quad (7F)_{16} = (127)_{10}$$

$$(5) \quad (2D3E)_{16} = (11582)_{10}$$

$$(6) \quad (F10E)_{16} = (61710)_{10}$$

**1-11** 简要比较原码、反码、补码等几种编码方法。

**解：**原码：将符号位用 0 和 1 表示，0 表示正，1 表示负，数的绝对值与符号一起编码，即所谓“符号-绝对值表示”的编码。

正数的反码和补码与原码表示相同。

负数的反码与原码有如下关系：符号位相同（仍用 1 表示），其余各位取反（0 变 1，1 变 0）。补码由该数反码的最末位加 1 求得。

# 第 2 章

## C++ 简单程序设计

2-1 C++ 语言有哪些主要特点和优点？

解：C++ 语言的主要特点表现在两个方面，一是全面兼容 C，二是支持面向对象的方法。C++ 是一个更好的 C，它保持了 C 的简洁、高效、接近汇编语言、具有良好的可读性和可移植性等特点，对 C 的类型系统进行了改革和扩充，因此 C++ 比 C 更安全，C++ 的编译系统能检查出更多的类型错误。C++ 语言最重要的特点是支持面向对象的方法。

2-2 下列标识符哪些是合法的？

Program、-page、\_lock、test2、3in1、@mail、A\_B\_C\_D

解：

Program、\_lock、test2、A\_B\_C\_D 是合法的标识符，其他的不是。

2-3 例 2.1 中每条语句的作用是什么？

```
#include<iostream.h>
void main(void)
{
    cout<<"Hello! \n";
    cout<<"Welcome to C++! \n";
}
```

解：

```
#include<iostream.h>           //指示编译器将文件 iostream.h 中的代码
                                //嵌入到本程序中该指令所在的地方
void main()                    //主函数名,void 表示函数没有返回值
{
    cout<<"Hello! \n";         //输出字符串"Hello!"到标准输出设备(显示器)上
    cout<<"Welcome to C++! \n"; //输出字符串"Welcome to c++!"
}
```

在屏幕输出如下：

```
Hello!
Welcome to C++!
```

**2-4 使用关键字 const 而不是 #define 命令的好处有哪些?**

**解:** const 定义的常量是有类型的,所以在使用它们时编译器可以检查类型错误;而且,这些常量在调试时是可见的。

**2-5 请写出 C++ 语句声明一个常量 PI, 值为 3.1416; 再声明一个浮点型变量 a, 把 PI 的值赋给 a。**

**解:**

```
const float PI=3.1416;
float a=PI;
```

**2-6 在下面的枚举类型中, BLUE 的值是多少?**

```
enum COLOR{WHITE, BLACK=100, RED, BLUE, GREEN=300};
```

**解:**

```
BLUE=102
```

**2-7 注释有什么作用? C++ 中有哪几种注释的方法? 他们之间有什么区别?**

**解:** 注释在程序中的作用是对程序进行注解和说明, 以便于阅读。编译系统在对源程序进行编译时不理睬注释部分, 因此注释对于程序的功能实现不起任何作用。而且由于编译时忽略注释部分, 所以注释内容不会增加最终产生的可执行程序的大小。适当地使用注释, 能够提高程序的可读性。在 C++ 中, 有两种给出注释的方法: 一种是沿用 C 语言的方法, 使用“/\*”和“\*/”括起注释文字。另一种方法是使用“//”, 从“//”开始, 直到它所在行的行尾, 所有字符都被作为注释处理。

**2-8 什么叫做表达式? x=5+7 是一个表达式吗? 它的值是多少?**

**解:** 任何一个用于计算值的公式都可称为表达式。x=5+7 是一个表达式, 它的值为 12。

**2-9 下列表达式的值是多少?**

- ① 201/4
- ② 201%4
- ③ 201/4.0

**解:**

- ① 50
- ② 1
- ③ 50.25

**2-10 执行完下列语句后, a、b、c 三个变量的值为多少?**

```
a=30;
b=a++;
c=++a;
```

**解:**

```
a:32;    b:30;    c:32;
```

2-11 在一个 for 循环中,可以初始化多个变量吗?如何实现?

解: 在 for 循环设置条件的第一个“;”前用“,”分隔不同的赋值表达式。

例如:

```
for (x=0, y=10; x<100; x++, y++)
```

2-12 执行完下列语句后,n 的值为多少?

```
int n;  
for (n=0; n<100; n++)
```

解: n 的值为 100。

2-13 写一条 for 语句,计数条件为 n 从 100 到 200,步长为 2;然后用 while 和 do...while 语句完成同样的循环。

解:

```
for 循环:  
for (int n=100; n<=200; n+=2);
```

```
while 循环:  
int n=100;  
while (n<=200)  
    n+=2;
```

```
do...while 循环:  
int n=100;  
do  
{  
    n+=2;  
} while(n<=200);
```

2-14 if (x=3)和 if(x==3)这两条语句的差别是什么?

解: 语句 if(x=3)把 3 赋给 x,赋值表达式的值为 true,作为 if 语句的条件;语句 if(x==3)首先判断 x 的值是否为 3,若相等,条件表达式的值为 true,否则为 false。

2-15 什么叫做作用域?什么叫做局部变量?什么叫做全局变量,如何使用全局变量?

解: 作用域是一个标识符在程序正文中有效的区域。局部变量一般来讲就是具有块作用域的变量;全局变量就是具有文件作用域的变量。

2-16 已知 x,y 两个变量,写一条简单的 if 语句,把较小的值赋给原本值较大的变量。

解:



```

if (x>y)
    x=y;
else
    y=x;
// y>x || y==x

```

2-17 修改下面这个程序中的错误,改正后它的运行结果是什么?

```

#include<iostream. h>
void main()
    int i
    int j;
    i=10;          /* 给 i 赋值
    j=20;          /* 给 j 赋值 */
cout<<"i+j="<<i+j; /* 输出结果 */
    return 0;
}

```

解:

改正:

```

#include<iostream. h>
int main()
{
    int i;
    int j;
    i=10;          // 给 i 赋值
    j=20;          /* 给 j 赋值 */
    cout<<"i+j="<<i+j; /* 输出结果 */
    return 0;
}

```

程序运行输出:

```
i+j=30
```

2-18 编写一个程序,运行时提示输入一个数字,再把这个数字显示出来。

解:

源程序:

```

#include<iostream. h>
int main()
{
    int i;
    cout<<"请输入一个数字:";
    cin >> i;
    cout<<"您输入一个数字是"<<i<<endl;
    return 0;
}

```