



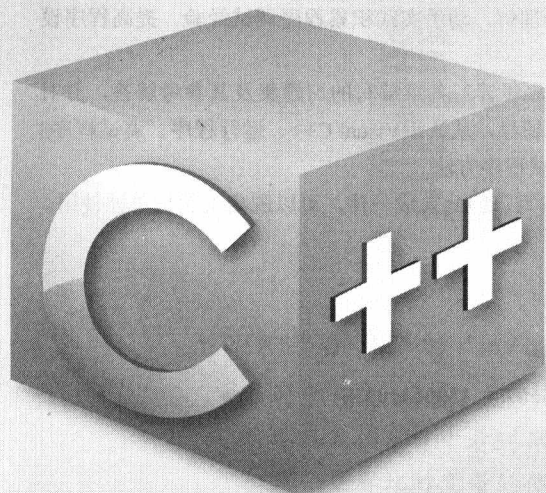
# C++ 面向对象 程序设计学习辅导

李春葆 董尚燕 余云霞 编著

- 习题补充教材，相辅相成
- 知识点归纳总结，强化理论基础
- 基本技能训练+综合提高练习，便于检验、  
巩固所学知识，提高面向对象的程序设计能力



清华大学出版社



# 面向对象 程序设计学习辅导

李春葆 董尚燕 余云霞 编著

清华大学出版社  
北京

## 内 容 简 介

本书作为《C++面向对象程序设计》(李春葆、董尚燕、余云霞编著,清华大学出版社,2007)一书配套的习题解答和上机学习指导,旨在加深对C++语言的理解,动手实践积累程序调试经验,提高程序设计技巧,从而达到举一反三的学习目的。

全书分为两部分,第 I 部分是为《C++面向对象程序设计》各章编写的习题集及其参考解答,并对相关知识点进行了系统的归纳和总结;第 II 部分为上机辅导,从认识Visual C++、运行程序、调试程序3个方面阐述Microsoft Visual C++ 6.0的集成开发环境和调试程序方法。

本书采用“学习要点+练习题及参考答案”的模式编写,因此自成一体,可以脱离主教材单独使用,也可作为计算机及相关专业本科、专科生的辅导教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

### 图书在版编目(CIP)数据

C++面向对象程序设计学习辅导 / 李春葆,董尚燕,余云霞编著.

—北京:清华大学出版社,2008

ISBN 978-7-302-16955-0

I. C… II. ①李…②董…③余… III. C语言—程序设计—高等学校—教学参考资料

IV. TP312

中国版本图书馆CIP数据核字(2008)第010863号

责任编辑:夏非彼 张楠

责任校对:贾淑媛

责任印制:孟凡玉

出版发行:清华大学出版社 地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn> 邮 编:100084

[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

社总机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印刷者:清华大学印刷厂

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:10.75 字 数:261千字

版 次:2008年2月第1版 印 次:2008年2月第1次印刷

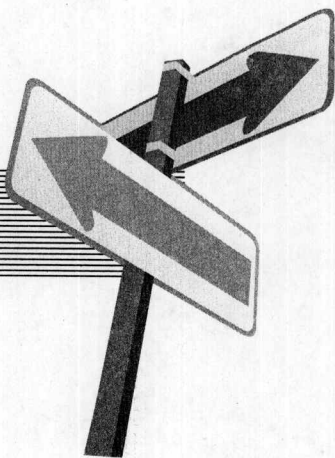
印 数:1~5000

定 价:16.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028636-01



# 前言



本书作为《C++面向对象程序设计》（李春葆、董尚燕、余云霞编著，清华大学出版社，2007）一书配套的习题解答和上机指导书。

全书分为两部分，第 I 部分是为《C++面向对象程序设计》各章编写的习题集及其参考解答，并对相关知识点进行了系统的归纳和总结；第 II 部分为上机指导，从认识 Visual C++、运行程序、调试程序 3 个方面阐述 Microsoft Visual C++ 6.0 的集成开发环境和调试程序方法。

本书采用“学习要点+练习题及参考答案”的模式编写，因此自成一体，可以脱离主教材单独使用，也可作为计算机及相关专业本科、专科生的辅导教材。

本书的编写得到武汉大学计算机学院的大力支持，特别是黄传河教授作为教学院长对教材的编写工作给予了积极的鼓励和不遗余力的推动，在此编者向其表示深深地感谢。

由于作者水平有限，书中难免存在疏漏和错误之处，恳请专家和广大读者批评指正。在学习过程中，如遇到疑难问题，可以通过以下方式与我们联系：[booksaga@126.com](mailto:booksaga@126.com)，也可以登录图格新知网站<http://www.booksaga.com>留言，我们将在第一时间给予答复！

编者

2007年10月

# 目 录

## 第 I 部分 练习题参考答案

第1章 C++基础	2
1.1 学习要点	2
1.2 练习题1及参考答案	2
1.3 上机实验题1及参考答案	10
第2章 类和对象（一）	12
2.1 学习要点	12
2.2 练习题2及参考答案	12
2.3 上机实验题2及参考答案	28
第3章 类和对象（二）	31
3.1 学习要点	31
3.2 练习题3及参考答案	31
3.3 上机实验题3及参考答案	37
第4章 友元	40
4.1 学习要点	40
4.2 练习题4及参考答案	40
4.3 上机实验题4及参考答案	45
第5章 运算符重载	47
5.1 学习要点	47
5.2 练习题5及参考答案	47
5.3 上机实验题5及参考答案	56
第6章 模板	60
6.1 学习要点	60
6.2 练习题6及参考答案	60
6.3 上机实验题6及参考答案	65
第7章 继承和派生	67
7.1 学习要点	67
7.2 练习题7及参考答案	67
7.3 上机实验题7及参考答案	82
第8章 虚函数和多态性	84
8.1 学习要点	84
8.2 练习题8及参考答案	84
8.3 上机实验题8及参考答案	91

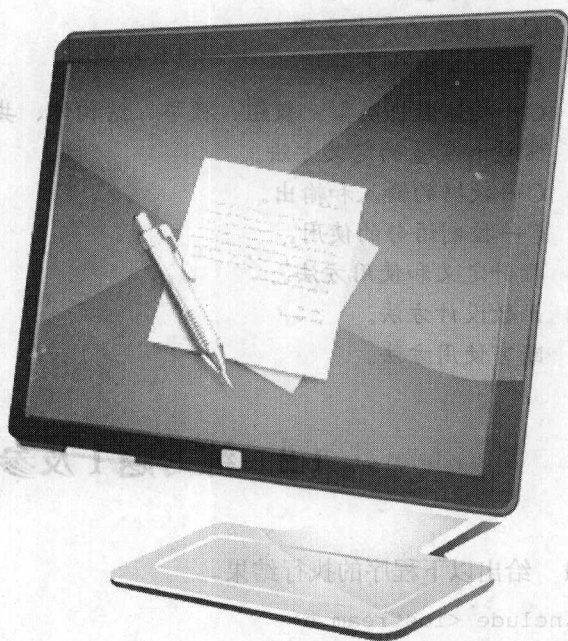


<b>第9章 C++流</b> .....	94
9.1 学习要点.....	94
9.2 练习题9及参考答案.....	94
9.3 上机实验题9及参考答案.....	104
<b>第10章 异常处理和名字空间</b> .....	107
10.1 学习要点.....	107
10.2 练习题10及参考答案.....	107
10.3 上机实验题10及参考答案.....	111
<b>第11章 C++标准模板库基础</b> .....	113
11.1 学习要点.....	113
11.2 练习题11及参考答案.....	113
11.3 上机实验题11及参考答案.....	120
<b>第12章 面向对象软件设计</b> .....	123
12.1 学习要点.....	123
12.2 练习题12及参考答案.....	123
12.3 上机实验题12及参考答案.....	124
<b>第II部分 C++上机指导</b>	
<b>第13章 认识Visual C++</b> .....	138
13.1 Visual C++的特点.....	138
13.2 Visual C++的安装.....	139
13.3 Visual C++的开发环境.....	145
13.3.1 集成开发环境窗口.....	145
13.3.2 工作区窗口.....	146
13.3.3 菜单栏和工具栏.....	148
13.3.4 编辑器.....	149
13.3.5 常用功能键及其意义.....	150
<b>第14章 运行程序</b> .....	151
14.1 运行单文件程序.....	151
14.2 运行多文件程序.....	153
<b>第15章 调试程序</b> .....	157
15.1 调试器的启动.....	157
15.2 调试器的使用.....	158
15.2.1 浏览调试器.....	158
15.2.2 设置与使用断点.....	159
15.2.3 查看和修改变量的值以及寄存器的内容.....	161
15.3 调试的方法.....	163
15.3.1 修正语法错误.....	164
15.3.2 调试异常.....	164

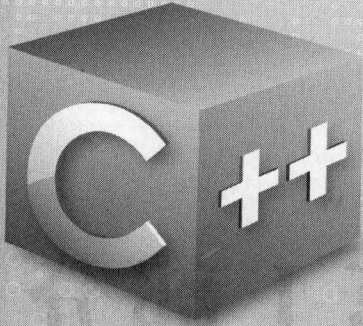


# 第 I 部分

## 练习题参考答案



# 第 1 章



## C++基础

### 1.1 学习要点

- 结构化程序设计和面向对象程序设计的特点。
- C++语言特点。
- C++程序基本结构。
- C++程序的开发步骤。
- C++内置数据类型。
- C++构造数据类型：数组、枚举、结构体、共用体。
- 常量和变量的定义方法。
- C++数据的输入和输出。
- C++控制语句的使用。
- 指针定义和使用方法。
- 函数设计方法。
- 断言使用方法。

### 1.2 练习题 1 及参考答案

1.1 给出以下程序的执行结果。

```
#include <iostream.h>
void main()
{   int i,j,k;
    for (i=1;i<=6;i++)
```



```

    {
        for (j=1;j<=20-2*i;j++)
            cout << " ";
        for (k=1;k<=i;k++)
            cout << i << " ";
        cout << endl;
    }
}

```

解：程序外循环执行6次，打印6行。对于第*i*次外循环，内循环输出数字*i*共*i*个。程序执行结果如下：

```

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6

```

### 1.2 给出以下程序的执行结果。

```

#include <iostream.h>
#include <iomanip.h>
void main()
{
    int i,j,k;
    for (i=1;i<=6;i++)
    {
        for (j=1;j<=20-3*i;j++)
            cout << " ";
        for (k=1;k<=i;k++)
            cout << setw(3) << k;
        for (k=i-1;k>0;k--)
            cout << setw(3) << k;
        cout << endl;
    }
}

```

解：程序外循环执行6次，打印6行。对于第*i*次外循环，先输出1~*i*的数字，再输出*i*-1~1的数字。程序执行结果如下：

```

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1

```

### 1.3 给出以下程序的执行结果。

```

#include <iostream.h>
void ast(int x,int y,int *cp,int *dp)

```



```

{
    *cp=x+y;
    *dp=x-y;
}
void main()
{
    int a,b,c,d;
    a=55;b=7;
    ast(a,b,&c,&d);
    cout << "c=" << c << ",d=" << d << endl;
}

```

解：函数 ast 中 cp 和 dp 形参是传地址方式，在 main 函数中调用 ast(a,b,&c,&d)后，c=a+b=55+7=62，d=55-7=48。程序输出为：c=62，d=48。

1.4 给出以下程序的执行结果。

```

#include <iostream.h>
int add(int a,int b,int c=0,int d=0)
{ return(a+b+c+d);
}
void main()
{ cout << add(1,2) << endl;
  cout << add(1,2,3) << endl;
  cout << add(1,2,3,4) << endl;
}

```

解：函数 add 带有两个默认参数值。程序执行结果如下：

3  
6  
10

1.5 给出以下程序的执行结果。

```

#include <iostream.h>
void main()
{
    static int a[]={2,6,10,14,18};
    static *ptr[]={&a[0],&a[1],&a[2],&a[3],&a[4]};
    int **p,i;
    for (i=0;i<5;i++)
        a[i]=a[i]/2+a[i];
    p=ptr;
    cout << *((p+2)) << ",";
    cout << *((++p)) << endl;
}

```

解：执行“static \*ptr[]={&a[0],&a[1],&a[2],&a[3],&a[4]};”语句后，使 ptr[0]指向 a[0]，…，ptr[4]指向 a[4]。执行 for 循环语句和 p=ptr 之后，存储结构如图 9.3 所示。  
\*((p+2))=\*ptr[2]=15，\*((++p))=\*ptr[1]=9。程序输出为：15，9

1.6 编写一个程序，从键盘输入 10 个学生的成绩，统计最高分、最低分和平均分。

解：使用一维数组 A 存储学生的成绩，然后用 for 循环进行判定求值。程序如下：

```

#include <iostream.h>
const int M=10;
void main()
{
    int i;
    float A[M],min,max,avg;
    cout << "输入十个学生的成绩:" << endl;
    for (i=0;i<M;i++)
    {
        cout << "第" << i+1 << "个学生成绩:";
        cin >> A[i];
    }
    min=max=avg=A[0];
    for (i=1;i<M;i++) //求最小值和最大值
    {
        if (min>A[i]) min=A[i];
        if (max<A[i]) max=A[i];
        avg=avg+A[i];
    }
    avg=avg/10;
    cout << "最高分:" << max << ",最低分:" << min << ",平均分:" << avg << endl;
}

```

1.7 设计一个函数 `strend(s,t)`, 假设字符串 `t` 出现在字符串 `s` 的尾端则返回 1, 否则返回 0。编写调用此函数的完整的程序。

**解:** 在函数 `strend(s,t)` 中, 求出 `s` 和 `t` 的长度分别为 `slen` 和 `tlen`。 `i` 从 0 开始循环, 若 `s[slen-tlen+i]` 不等于 `t[i]`, 则退出循环。最后如果 `i` 等于 `tlen`, 表示 `t` 比较完毕, 则 `t` 出现在字符串 `s` 的尾端, 返回 1, 否则返回 0。程序如下:

```

#include <iostream.h>
const int Max=100;
bool strend(char s[],char t[])
{
    int i,slen=0,tlen=0;
    while (s[slen++]);
    slen--;
    while (t[tlen++]);
    tlen--;
    if (tlen>slen) return false;
    i=0;
    for (i=0;t[i]!='\0' && t[i]==s[slen-tlen+i];i++);
    if (i==tlen)
        return true;
    else
        return false;
}
void main()
{
    char s[Max],t[Max];
    cout << "s:";
    cin.getline(s,Max);
    cout << "t:";
    cin.getline(t,Max);
}

```







```

if (strend(s,t))
    cout << "t 出现在 s 的尾端" << endl;
else
    cout << "t 不出现在 s 的尾端" << endl;
}

```

### 1.8 编写一个程序，将十进制整数 n 转换成二进制数。

**解：**采用辗转相除法进行十进制数到二进制数的转换，用一个长度为 16 的一维数组存放该二进制数。程序如下：

```

#include <iostream.h>
void trans(int n,int b[]) //该函数用于十进制整数 n 转换成二进制数
{
    int i=0; //从低位到高位存放二进制数的各位
    while (n!=0)
    {
        b[i]=n%2;
        i++;
        n=n/2;
    }
}
void main()
{
    int b[16],n,i;
    for (i=0;i<=15;i++)
        b[i]=0;
    cout << "n=";
    cin >> n;
    trans(n,b);
    cout << "二进制数为:";
    for (i=15;i>=0;i--) //从高位到低位输出二进制数的各位
        cout << b[i];
    cout << endl;
}

```

### 1.9 编写一个程序，其功能是将用户输入的十六进制数转换成与它等价的十进制整数值。十六进制允许的字符是从 0~9、a~f 或 A~F。

**解：**main 函数用于收集用户输入的十六进制数字的字符串，并在输入时判别它们是否为十六进制字符。设计一个函数 htoi，它用于进行进制转换工作。程序如下：

```

#include <iostream.h>
const int MAXLINE=100;
int htoi(char s[]);
void main()
{
    int m;
    char t[MAXLINE];
    cout << "十六进制数字串:";
    cin >> t;
    m=htoi(t);
    cout << m << endl;
}

```

```
int htoi(char s[]) //将s转换成十进制数
```

```
{
    int i,n=0;
    for (i=0;s[i]!='\0';i++)
    {
        if (s[i]>='0' && s[i]<='9')
            n=n*16+s[i]-'0';
        if (s[i]>='a' && s[i]<='f')
            n=n*16+s[i]-'a'+10;
        if (s[i]>='A' && s[i]<='F')
            n=n*16+s[i]-'A'+10;
    }
    return(n);
}
```

1.10 编写一个函数 fun(s1,s2)，将字符串 s1 中所有出现在字符串 s2 中的字符删去。

**解：**设计一个函数 in(ch,str)，用于判断字符 ch 是否在字符串 str 中，若在，则该函数返回 1，否则返回 0；在 fun(s1,s2)函数中，从前向后扫描字符串 s1，调用 in 函数判断 s1 的当前字符是否在字符串 s2 中，并用 k 记录下在字符串 s2 中的元素个数。当 k 大于 0 时，则将其后的所有元素前移 k 个位置。这样只需扫描一趟，所以算法效率高。程序如下：

```
#include <iostream.h>
const int Max=250;
int in(char ch,char str[]) //判断字符ch是否在字符串str中
{
    int i=0;
    while (str[i]!='\0')
    {
        if (str[i]==ch)
            return 1;
        i++;
    }
    return 0;
}
void fun(char s1[],char s2[])
{
    int i=0,k=0;
    while (s1[i]!='\0')
    {
        if (in(s1[i],s2)==1)
            k++;
        else
            s1[i-k]=s1[i];
        i++;
    }
    s1[i-k]='\0'; //s1缩短，置字符串结束符
}
void main()
{
    char s1[Max],s2[Max];
    cout << "s1:";
    cin.getline(s1,Max);
    cout << "s2:";
```





```

cin.getline(s2,Max);
fun(s1,s2);
cout << "s1:" << s1 << endl;
cout << "s2:" << s2 << endl;
}

```

1.11 编程实现将字符串中最短的单词输出，在主函数中输入字符串，编写一个函数完成最短单词的查找。例如：

输入单词串：Happy new year.

最短单词：new。

**解：**假设单词之间用空格分隔（一个或多个空格）。findshort(s)函数完成 s 串中最短单词的查找。为了简明，先在 s 末尾添加一个空格，并过滤掉过余的空格（使单词之间仅有一个空格分隔）。用 s2 从 s 中提取当前单词，s1 保存最短单词，将第一个提取的单词放入 s1 中，对于后面提取的单词 s2，比较长度，总是将较短的单词放在 s1 中。最后返回 s1 到主函数并输出。程序如下：

```

#include <iostream.h>
const int Max=100;
char *findshort(char s[])
{
    static char s1[Max]; //其地址要返回,所以设计为静态变量
    char s2[Max];
    int i=0,j,len1=0,len2=0;
    while (s[i++]!='\0'); //为了简明,在 s 末尾添加一个空格
    s[i-1]=' ';
    s[i]='\0';
    i=0;
    while (s[i]!='\0')
    {
        if (s[i]==' ' && s[i+1]!='\0' && s[i+1]!=' ') //跳过多余的空格
        {
            i++;
            continue;
        }
        if (s[i]!=' ') //提取一个单词到 s2 中
            s2[len2++]=s[i];
        else if (len1==0) //给 s1 置初值
        {
            len1=0;
            for (j=0;j<len2;j++) //将 s2 复制到 s1 中
                s1[len1++]=s2[j];
            s1[len1]='\0';
            len2=0;
        }
        else if (len1>len2)
        {
            len1=0;
            for (j=0;j<len2;j++) //将 s2 复制到 s1 中
                s1[len1++]=s2[j];
            s1[len1]='\0';
            len2=0;
        }
    }
}

```



```

    }
    i++;
}
return s1;
}
void main()
{
    char s[Max];
    cout << "输入单词串:";
    cin.getline(s,Max);
    cout << "最短单词:" << findshort(s) << endl;
}

```

1.12 一个素数，当它的数字位置对换以后仍为素数，这样的数称为绝对素数。编写一个程序，求出所有的两位绝对素数。

**解：** $n$  从 11~97 进行循环，如果调用  $\text{prime}(n)$  和  $\text{prime}(\text{inv}(n))$  函数均返回真，则  $n$  是绝对素数，打印它。其中， $\text{prime}(n)$  是判定  $n$  是否为素数，判定  $n$  为素数的原理是：如果  $n$  能够被 2 整除或  $3 \sim \sqrt{\text{prime}(n)}$  之间的某个奇数整除，则  $n$  不是素数，否则  $n$  为素数。 $\text{inv}(a)$  用于求两位数的数字位置对换后的数。程序如下：

```

#include <iostream.h>
#include <math.h>
int inv(int a);
prime(int a)
{
    int b,k;
    if (a==2)
        return(1);
    else if (a%2==0)
        return(0);
    else
    {
        b=1;
        k=3;
        while (k<=sqrt(a) && b)
        {
            if (a%k==0) b=0;
            k=k+2;
        }
        return(b);
    }
}

```

由 int inv(int a) //返回两位的逆数，如参数为 25 时返回 52

```

{
    int i,j;
    i=a/10;
    j=a%10;
    return(j*10+i);
}
void main()
{

```

```

    int n;
    for (n=11;n<=97;n++)

```



```

        if (prime(n) && prime(inv(n)))
            cout << n << " ";
        cout << endl;
    }
}

```

1.13 编写一个程序，执行通配符'?'的匹配，该通配符可以与任一个字符匹配成功。

**解：**从头开始扫描字符串 str 的每个字符，当当前字符等于子串的字符或者子串的当前比较的字符是'?'时，子串指针向前推进，当子串指针为'\0'时比较完毕并返回对应位置；否则重新从 str 串的当前位置与子串进行比较。当未找到匹配的且 str 结束时返回-1。程序如下：

```

#include <iostream.h>
int pattern_index(char substr[],char str[]);
void main()
{
    char *sstr,*mstr;
    int index;
    sstr="???gram";
    mstr="this program return index of substring";
    index=pattern_index(sstr,mstr);
    if (index!=0)
        cout << "子串起始位置:" << index << endl;
    else
        cout << "匹配不成功." << endl;
}
int pattern_index(char substr[],char str[])
{
    int i,j,k;
    for (i=0;str[i];i++)
    {
        for (j=i,k=0;(str[j]==substr[k]) || (substr[k]=='?');j++,k++);
        if (!substr[k+1])
            return(i);
    }
    return(0);
}

```

### 1.3 上机实验题 1 及参考答案

编写一个程序，将用户输入的由数字字符和非数字字符组成的字符串中的数字提取出来，例如输入"asd123rt456,fg789"，则产生的数字分别是 123、456 和 789。

**解：**先用 cin.getline 成员函数获取用户输入的一行字符，并存放在 line 字符数组中；然后扫描 line 的各个字符，判定字符的类型是否为数字型，并将连续数字经过转换后放入数组 a 中。这一过程直到 line 结束为止。最后输出 a 数组的结果。程序如下：

```

#include <iostream.h>
#include <iomanip.h>
#include <stdlib.h>

```

```

const int Max=256;
void main()
{
    char line[Max],b[20],*ptrl,*ptrb;
    int n=0,j,a[50];
    cout << "输入串:";
    cin.getline(line,Max);
    ptrl=line;
    while (*ptrl!='\0')
    {
        ptrb=b;
        for (j=0;*ptrl>='0' && *ptrl<='9';ptrl++,ptrb++,j++)
            *ptrb=*ptrl;
        if (j)
        {
            *ptrb='\0';
            *(a+n)=atoi(b);    //atoi 函数将数字串 b 转换为整数
            n++;
        }
        if (*ptrl!='\0') ptrl++;
    }
    cout << "输出结果:" << endl;
    for (j=0;j<n;j++)
        cout << setw(10) << *(a+j) << endl;
}

```

程序的执行结果如下:

输入串:asdl23rt456,fg789↵

输出结果:

123  
456  
789

