

21

世纪高等学校规划教材

C++

面向对象程序设计 习题与实验指导

康 丽 主 编
孙龙清 郑立华 马 钦 副主编



中国电力出版社

<http://jc.cepp.com.cn>

C++ 面向对象程序设计 习题与实验指导

主 编 康 丽

副主编 孙龙清 郑立华 马 钦

参 编 张瑞玲 王莲芝 程新荣 杨 璐

徐红升 王玉娟 冀荣华

主 审 宗绪锋



中国电力出版社

<http://jc.cepp.com.cn>

内 容 提 要

本书是《C++面向对象程序设计简明教程》的配套教材,所选内容有教材中各章习题与解答、上机实验与参考答案、C++课程设计指导和计算机等级考试 C++模拟试题与答案等,在加深学生对 C++内容的理解、掌握程序设计技巧和积累程序调试经验、培养动手能力等方面能起到重要的作用。

本书突出了 C++面向对象程序设计的知识要点,与课堂教学紧密配合,可作为学习 C++语言的配套教材,可作为上机实验、课程设计的独立教材,也可作为计算机爱好者和程序员在 C++编程方面的参考用书。

图书在版编目(CIP)数据

C++面向对象程序设计习题与实验指导 / 康丽主编. —北京: 中国电力出版社, 2010

21 世纪高等学校规划教材

ISBN 978-7-5083-9935-5

I. ①C… II. ①康… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2009) 第 239637 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

北京丰源印刷厂印刷

各地新华书店经售

*

2010 年 2 月第一版 2010 年 2 月北京第一次印刷

787 毫米×1092 毫米 16 开本 11.25 印张 268 千字

印数 0001—3000 册 定价 18.00 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前言

在面向对象语言 C++ 的学习中,做作业和上机实验既可以加深对面向对象程序设计思想和概念的理解,同时也是掌握 C++ 和面向对象方法的基础知识、提高计算机程序设计技巧和积累编程经验的基本途径,所以我们编写了这本书。

本书通过细致的讲解、具体的实例及其运行结果的显示,让读者能够快速地掌握 C++ 编程的基本方法,加快学习步伐,快速上手。本书涵盖了《C++ 面向对象程序设计简明教程》中各个章节的全部知识点。其中第 1 章给出了《C++ 面向对象程序设计简明教程》一书中所有习题的参考答案,第 2、3 章给出了上机实验的目的、内容要求和上机实验的实验结果及参考答案,第 4 章是 C++ 课程设计指导,包括课程设计要求、课程设计预备知识、两个课程设计案例(高等院校人员管理系统和飞机航班票务系统)和 4 个课程设计题目,第 5 章包括 3 套计算机等级考试 C++ 笔试模拟试题与答案。

本书由康丽担任主编,由马钦、郑立华、孙龙清和其他老师参加编写。其中上机实验项目与实验参考答案由康丽和孙龙清编写,C++ 课程设计指导由马钦编写,计算机等级考试模拟试题与答案由郑立华编写,各章习题解答由原教材各章编写老师王莲芝、程新荣、杨璐、徐红升、王玉娟、冀荣华等共同编写,全书由宗绪锋审稿。

由于作者水平和经验有限,时间仓促,书中难免存在错误和不足之处,恳请读者不吝指出,便于在以后的版本中进行改进。具体联系方式:康丽 中国农业大学计算机系,E-mail: kangli.cau@gmail.com。

编者
2009 年 11 月

目

录

前言

第1章 习题解答	1
习题1 计算机与 C++编程	1
习题2 基本数据类型和表达式	2
习题3 程序的基本控制结构与算法	5
习题4 函数与程序结构	7
习题5 构造数据类型	13
习题6 类与对象	20
习题7 继承与派生	25
习题8 多态性	26
习题9 模板	30
习题10 流类库与输入/输出	33
习题11 异常处理	37
第2章 上机实验	38
实验1 Visual C++ 6.0 开发环境学习和程序设计基础	38
实验2 C++程序设计基础	41
实验3 函数	44
实验4 构造数据类型和类	48
实验5 类与对象	54
实验6 继承与派生	59
实验7 多态	63
实验8 虚函数与多态	67
实验9 模板	74
实验10 I/O 流类库	76
实验11 异常处理	78
第3章 上机实验参考答案	84
实验1 Visual C++ 6.0 开发环境学习和程序设计基础	84
实验2 C++程序设计基础	85
实验3 函数	88
实验4 构造数据类型和类	91
实验5 类与对象	93
实验6 继承与派生	99
实验7 多态	101
实验8 虚函数与多态	104

实验 9 模板	110
实验 10 I/O 流类库	114
实验 11 异常处理	115
第 4 章 C++课程设计指导	117
4.1 课程设计要求	117
4.2 课程设计预备知识	118
4.3 课程设计案例	124
题目一：高等院校人员管理系统	124
题目二：飞机航班票务系统	137
4.4 课程设计题目	144
题目一：歌曲大奖赛管理系统	144
题目二：音像出租店管理系统	145
题目三：大学生社会实践管理系统	146
题目四：三子棋小游戏系统	147
第 5 章 计算机等级考试二级 C++笔试模拟试题与答案	148
计算机等级考试二级 C++笔试模拟试题 (1)	148
计算机等级考试二级 C++笔试模拟试题 (1) 答案	155
计算机等级考试二级 C++笔试模拟试题 (2)	156
计算机等级考试二级 C++笔试模拟试题 (2) 答案	162
计算机等级考试二级 C++笔试模拟试题 (3)	163
计算机等级考试二级 C++笔试模拟试题 (3) 答案	170
参考文献	171

第1章 习题解答

习题1 计算机与 C++编程

一、填空题

1. 从某种意义上讲,一台计算机的性能主要由硬件决定,而它的功能则主要是由软件来提供。

2. 冯·诺依曼计算机的工作模型:待执行的程序从外存装入到内存中,CPU 从内存中逐条地取程序中的指令执行;程序执行中所需要的数据从内存或从外设中获得,程序执行中产生的中间结果保存在内存中,程序的执行结果通过外设输出。

3. 计算机的工作是用程序来控制的。程序是指令的集合。指令是计算机可以识别的命令。

4. C++中使用标识符定义的实体包括常量、变量、函数、类、对象、标号等。

5. C++中的单行注释是以符号“//”开始到本行结束。多行注释是以符号“/*”开始到符号“*/”结束。

二、简答题

1. 高级语言与低级语言有何区别?

答:高级语言与低级语言的主要区别在于它们与计算机指令系统和人们解决问题所采用的描述语言(如数学语言)的接近程度不同。高级语言屏蔽了机器的细节,提高了语言的抽象层次,程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这使得在书写程序时可以联系到程序所描述的具体事物。而低级语言依赖于具体的计算机系统,在人类的自然语言和计算机编程语言之间存在着巨大的鸿沟,因为它的抽象层次太低,程序员需要考虑大量的机器细节。

2. 举例说明高级语言如何让计算机得到理解并执行?

答:因为机器只知道根据机器语言指令来执行程序,高级语言程序必须经过翻译变成机器语言程序,这个工作由翻译程序完成。将高级语言程序翻译成机器语言程序的翻译程序有编译器和解释器两种。编译器(Compiler)是将一种高级语言完整地翻译成另一种低级语言的翻译程序,编译器的重要性在于它使得多数计算机用户不必考虑与机器有关的烦琐细节,使程序员和程序设计专家独立于机器而只专注于编程。解释器(Interpreter)是将源程序逐句读入,逐句执行,直到程序结束。即对源程序边解释边执行,不产生目标程序。在调试程序或者教学中,需要随时检验程序的执行结果或需要找出程序中的问题时,采用高级语言的解释执行方式比较方便。例如,C++和 Java 语言采用编译和解释混合的方式。

3. 目标程序与可执行程序的差异?

答:C++程序的开发通常要经过编辑、编译、连接、运行调试这4个步骤。其中,编辑是将C++源程序输入到计算机中,并作为文件(称为源程序)保存到外存(如硬盘)中,C++源程序的文件名通常为*.cpp和*.h。编译是将C++源程序编译生成目标文件保存到外存,目标文件的文件名通常为*.obj。连接是将多个目标文件以及程序中用到的一些系统功能所在的

目标文件，即库中的某些文件连在一起，生成一个后缀为*.exe的可执行文件。最后，把某个可执行文件装入内存运行并给出结果。在编译、连接、运行时，都有可能碰到程序的错误，所以有可能需要回到编辑中进行改正。由此可知，目标程序是对源程序编译后生成的程序，有可能需要连接库文件才形成可执行程序。

三、判断题

1. 一般来说，解释执行比编译执行效率要高。 (×)
2. 程序设计的一般步骤是：需求分析、系统设计、编程实现、测试与调试、运行与维护。 (√)
3. C++语言的突出优点是支持面向对象的特征，所以编程人员只能采用面向对象方法编写程序。 (×)
4. C++字符集是由 26 个大小写英文字母和 10 个数字构成的。 (×)

习题 2 基本数据类型和表达式

一、选择题

1. 下列哪个是 C++语言的有效标识符 (A)。
A. _No1 B. No.1 C. 12345 D. int
2. 设有定义 int x; float v;，则 10+x+v 值的数据类型是 (C)。
A. int B. double C. float D. 不确定
3. 设 ch 是 char 型变量，其值为 A，且有下列的表达式：
ch=(ch>='A'&&ch<='Z')?(ch+32):ch 的值是 (B)。
A. A B. a C. Z D. z
4. 若定义 int k=7, x=12;，则值为 3 的表达式是 (D)。
A. x%=(k%=5) B. x%=(k-k%5) C. x%=k-k%5 D. (x%=k)-(k%=5)
5. 错误的转义字符是 (A)。
A. '\091' B. '\\\' C. '\0' D. '\"'
6. 在 C 语言中，错误的常数是 (A)。
A. 1E+0.0 B. 5 C. 0xaf D. 0L
7. 下面运算符优先级最高的是 (C)。
A. <= B. = C. % D. &&
8. C++语言提供的不合法的数据类型关键字是 (C)。
A. double B. short C. integer D. char
9. C++语言中，运算对象必须是整型数的运算符是 (A)。
A. % B. \ C. %和\ D. **
10. 表达式：10!=9 的值是 (D)。
A. true B. 非零值 C. 0 D. 1
11. 表示关系 $x \leq y \leq z$ 的 C++语言表达式为 (A)。
A. (x<=y)&&(y<=z) B. (x<=y)AND(y<=z)
C. (x<=y<=z) D. (x<=y)&(y<=z)

12. 若变量 a 是 `int` 类型, 并执行了语句: $a='A'+1.6;$, 则正确的叙述是 (D)。
- A. a 的值是字符 C B. a 的值是浮点型
C. 不允许字符型和浮点型相加 D. a 的值是字符 'A' 的 ASCII 值加上 1。
13. 若已定义 x 和 y 为 `double` 类型, 则表达式: $x=1, y=x+3/2$ 的值是 (C)。
- A. 1 B. 2 C. 2.0 D. 2.5
14. 若有以下定义: `char a; int b; float c; double d;` 则表达式 $a*b+d-c$ 值的类型为 (D)。
- A. `float` B. `int` C. `char` D. `double`
15. 设 x 和 y 均为 `int` 型变量, 则以下语句: $x+=y; y=x-y; x-=y;$ 的功能是 (D)。
- A. 把 x 和 y 按从大到小排列 B. 把 x 和 y 按从小到大排列
C. 无确定结果 D. 交换 x 和 y 中的值
16. 设有说明语句: `char a='\072';` 则变量 a (A)。
- A. 包含 1 个字符 B. 包含 2 个字符
C. 包含 3 个字符 D. 说明不合法
17. 字符 (`char`) 型数据在微机内存中的存储形式是 (D)。
- A. 反码 B. 补码 C. EBCDIC 码 D. ASCII 码

二、写出下列表达式的值

(1) `int a=1, b=4, c=2;`
`double m=10.5, n=4.0, k;`
`k=(a+b)/c+sqrt(n)*1.2/c+m;`

表达式的值为: 13.7

(2) `double x=2.5, y=4.7;`
`int a=7;`
`x+a%3*(int)(x+y)%2/4`

表达式的值为: 2.5

(3) `int a, b;`
`a=2, b=5, a++, b++, a+b;`

表达式的值为: 9

(4) `int a=10, b=20, c;`
`c=a>b?a+b:a-b;`

表达式的值为: -10

三、编程题

1. 编写四种不同的 “+” 语句, 对整数变量 x 加 1。

解: `x=x+1;`
`x+=1;`
`++x;`
`x++;`

2. 编写完成下列任务的 C++ 语句:

- (1) 将 x 和 y 的和赋给 z , 并在计算之后将 x 的值加 1。
(2) 测试变量 `count` 的值是否大于 10。如果是, 则打印 “Count is greater than 10”。
(3) 将变量 x 减 1, 然后将 `total` 变量的值减去 x 。

(4) 计算 q 除以 $divisor$ 的余数, 并将结果赋给 q 。用两种方法编写这个语句。

解: (1) $z=(x++)+y;$

(2) $\text{if}(\text{count}>10)$

$\text{cout} << \text{"Count is greater than 10"} << \text{endl};$

(3) $\text{total}----x;$

(4) $q\%=divisor;$

$q=q\%divisor;$

3. 编写完成下列任务的 C++ 语句:

(1) 将变量 sum 和 x 声明为 int 类型。

(2) 将变量 x 初始化为 1。

(3) 将变量 sum 初始化为 0。

(4) 将 x 加进 sum 中, 并将结果赋给 sum 。

(5) 打印 "The sum is:" 加上 sum 的值。

解: (1) $\text{int } sum, x;$

(2) $x=1;$

(3) $sum=0;$

(4) $sum+=x;$ 或 $sum = sum + x;$

(5) $\text{cout} << \text{"The sum is:"} << sum << \text{endl};$

4. 确定计算完成后每个变量的值。假设每个语句开始执行时, 所有变量为整数值 5。

(1) $\text{product}*=x++;$

(2) $\text{quotient} /=++x;$

解: (1) $\text{product} = 25, x = 6;$

(2) $\text{quotient} = 0, x = 6;$

5. 编写一条 C++ 语句:

(1) 用 cin 和 $>>$ 输入整型变量 x 。

(2) 用 cin 和 $>>$ 输入整型变量 y 。

(3) 将整型变量 i 初始化为 1。

(4) 将整型变量 power 初始化为 1。

(5) 将变量 power 乘以 x 并将结果赋给 power 。

(6) 将变量 i 加 1。

(7) 测试 i 是否小于或等于 y 。

(8) 用 cout 和 $<<$ 输出整型变量 power 。

解: (1) $\text{cin} >> x;$

(2) $\text{cin} >> y;$

(3) $i=1;$

(4) $\text{power}=1;$

(5) $\text{power}*=x;$ 或 $\text{power}=\text{power}*x;$

(6) $i++;$

(7) $\text{if}(i\leq y)$

(8) `cout << power << endl;`

四、简答题

1. 什么是符号常量？符号常量的优点是什么？

答：符号常量是用一个标识符命名的常量。符号常量的优点有以下两点。

(1) 可以使程序更清晰易读。

(2) 程序更易修改。

2. 什么是变量？变量定义的作用是什么？

答：变量是指程序执行过程中，其值可以改变的量。变量定义的作用是指出变量的类型和变量名，还可以为变量提供一个初值。

3. 什么是表达式？

答：表达式是由操作符，操作数以及圆括号所组成的运算式，其中，操作符是用于实现操作的运算符号，如关系、逻辑、算术、函数调用、数组元素选择、结构/类成员选择、取地址操作符等。操作数是操作符的运算所需要的数据，它们可以是常量、变量或函数调用，也可以是用圆括号括起来的表达式。单独的常量或变量构成了表达式的特例，称为基本表达式，从语法上讲，用圆括号括起来的表达式也属于基本表达式。

4. 操作符的优先级和结合性分别指的是什么？

答：操作符的优先级是指相邻的两个操作符的优先运算顺序。规定了优先级高的先运算，如果相邻的两个操作符具有相同的优先级，则需根据操作符的结合性来决定先计算谁。操作符的结合性又分为左结合和右结合。左结合指先左后右计算；右结合指先右后左计算。

习题 3 程序的基本控制结构与算法

一、填空题

1. 下列程序段循环的次数是 9 次。

```
int x=10;
while(--x) {}
```

2. 下面 for 语句的循环次数为 7 次。

```
for(int i=0,x=0; !x && i<=6;i++)
```

3. 下面程序将输出字符 '*' 的个数是 99 个。

```
#include <iostream.h>
void main()
{
    int i=100;
    while(1)
    {
        i--;
        if(i==0) break;
        cout<<"*";
    }
}
```

4. 下面程序的作用是 输出一个由长 x 宽 y 个 “@” 构成的矩形图案。

```
#include <iostream.h>
int main( )
{
    int x,y;
    cout<<"Enter two integers in the range 1-20:";
    cin>>x>>y;
    for(int i=1;i<=y;i++)
    {
        for(int j=1;j<=x;j++)
            cout<<'@';
        cout<<endl;
    }
    return 0;
}
```

二、编程题

1. 编写一个程序，输出 13~100 之间的全部素数。要求素数之间以空格分隔。

参考程序如下：

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int n;                //测试 n 是不是素数
    int i;                //循环计数器
    int is_prime;         //布尔标志
    //假定数字是素数
    is_prime=true;
    //从键盘获取一个数字
    for(n=13;n<=100;n=n+2)
    //用 2~sqrt(n) 的所有整数来除它,看是否能整除
    {
        for(i=2;i<=sqrt((double)n);i++)
            if(n%i==0)
                is_prime=false;
        //打印结果
        if(is_prime)
            cout<<n<<" ";
        is_prime=true;
    }
    return 0;
}
```

2. 编写一个程序，输入 10 个数，打印其中最大数。

参考程序如下：

```
#include <iostream>
using namespace std;
int main()
```

```

{
    int d,max=0;
    int i=0;
    while(i<=10)
    { cin>>d;
      if(d>max)
        max=d;
      i++;
    }
    cout<<" max= " <<max;
}

```

3. 编写一个程序，输出如图 3-12 所示的效果。

参考程序如下：

```

#include <iostream>
using namespace std;
int main()
{ int i,j,n=4;
  for(i=1;i<=n;i++)                //输出前 4 行图案
  { for(j=1;j<=30+n-2*i;j++)
    { cout<<' ';                    //在图案左侧空 30 列
      for(j=1; j<=2*i-1 ;j++)
        cout<<'*'<<' ';
      cout<<endl;
    }
  }
  for(i=1;i<=n-1;i++)              //输出后 3 行图案
  { for(j=1;j<=30+i*2-n;j++)
    { cout<<' ';                    //在图案左侧空 30 列
      for(j=1; j<=7-2*i ;j++)
        cout<<'*'<<' ';
      cout<<endl;
    }
  }
}

```

```

      *
    * * *
  * * * * *
* * * * * *
  * * * * *
    * * *
      *

```

图 3-12 输出效果图

4. 编写一个程序，使用 for 循环来打印 m1~m2 的所有数字，m1 和 m2 由用户输入。

参考程序如下：

```

#include <iostream>
using namespace std;
int main()
{ int i,j,m1,m2;
  cout<<"m1,m2=(0,1,...,9)m1<m2"<<endl;
  cin>>m1>>m2;
  for(i=m1;i<=m2;i++)
    cout<<i<<' ';
}

```

习题 4 函数与程序结构

一、填空题

1. 函数的返回值是通过函数中的 return 语句获得的，因此，当函数有返回值时，函数体内要有该语句。

2. 根据变量定义位置的不同,可分为 局部变量 和 全局变量。
3. 一个程序将操作系统分配给其运行的内存块分为 代码区、全局数据区、堆区 和 栈区。
4. C++提供的编译预处理功能主要有: 宏定义命令、文件包含命令 及条件编译命令。
5. 递归调用的执行过程分为 递推过程 和 回归过程。

二、简答题

1. 函数的返回类型有几种? 它们的含义和使用方法是什么?

答: 函数的返回值有以下三种情形。

(1) 返回 void 类型。如果函数无值返回, 应说明为 void 类型。例如:

```
void print( ) {cout<<"Hello World !";}
void show( ) {cout<<"Wonderful C++!";}
```

函数仅需完成打印和显示工作, 不需返回任何数据, 这类函数调用一般形成一个函数调用语句。未作类型说明的函数, 系统认为是 int 类型函数, 返回一整型值。

(2) 返回数值类型。最常见的函数是返回一个数值的函数。例如:

```
int add(int a, int b);
float cuberoot(float x);
```

这类函数的调用表达式可以出现在表达式语句中。当函数要返回的数值不止一个时, 情况比较复杂, 一般它可以以结构或类的形式, 也可以以数组或对象指针的形式实现。

(3) 返回引用类型。C++语言提供的引用返回概念是其特有的一种很强的功能, 当函数定义中把该函数说明为某类型的引用类型时, 该函数调用后返回的不仅是值, 而且包含返回值的变量(或对象)。

2. C++中参数传递机制有几种? 分别简述这些机制。

答: 参数传递机制有值调用和引用调用。

(1) 值调用。值调用实参与形参的结合过程: 调用函数时, 系统为形式参数分配新的存储单元, 将实参的值赋给形参后, 被调函数中的操作是在形参的存储单元中进行的, 当函数调用结束时释放形参所占的存储单元。因此, 在函数中对形参值的任何修改都不会影响到实参的值。

(2) 引用调用。引用是指在程序中为一个变量取一个别名, 以便在不同的情况下使用, 对引用的操作也就是对目标变量的操作, 它们代表着同一存储单元。因此, 当形参被定义为引用类型, 则对形参的任何操作都意味着对相应实参的操作, 实参的值将随函数体内对形参的改变而改变。

3. 标识符作用域的分类, 并简述这些分类。

答: 标识符作用域共分成块作用域、文件作用域、函数声明(原型)作用域、函数作用域、类作用域五种。

(1) 块作用域。块是用花括号括起来的一段程序。在块内说明的变量具有块作用域, 其作用域是从变量说明处到块的结束处(即块的右花括号处)。

(2) 文件作用域。文件作用域也称全局作用域, 定义在所有函数之外的标识符, 具有文件作用域, 作用域为从定义处到整个源文件结束。文件中定义的全局变量和函数都具有文件作用域。

(3) 函数声明作用域。在进行函数声明时,形参作用域只在声明中,即作用域结束于右括号。正是由于形参不能被程序的其他地方引用,所以通常只要声明形参个数和类型,形参名可省略。

(4) 函数作用域。函数中定义的变量,包括形参和函数体中定义的局部变量,作用域都在该函数内,也称作函数域。

(5) 类作用域。类作用域是指类定义范围(包括类的声明部分和相应成员函数实现整个范围)。在该范围内,类的成员函数对数据成员有无限访问权限。

4. 内联函数的概念是什么?为什么要引入内联函数?

答:内联函数也称为内嵌函数,在使用内联函数时,C++编译器用函数体中的代码插入到调用该函数的语句之处,在程序运行时不再进行函数调用,从而消除函数调用时的系统开销,提高程序执行效率。

函数的使用可以减少程序的目标代码,实现程序代码共享,为编程带来方便。但在程序执行过程中调用函数时,系统要将程序当前的一些状态信息保存到堆栈中,同时转到被调函数的代码处去执行函数体语句,这些参数保存与传递的过程中系统需要时间和空间的开销,使程序执行效率降低。特别是对于那些代码较短而又频繁调用的函数,这个问题尤为严重。为了解决这一问题,C++引入了内联函数。

5. 什么是函数重载,并简述重载函数的目的以及参数满足的条件。

答:在C++中,两个或两个以上的函数可以重名,但要求函数的参数说明不同,如参数的类型不同,参数的个数不同等。这种共享同名函数称为函数重载。重载函数的意义就在于,可以用同一个函数名字访问多个相关函数,编译器能够根据参数的具体情况决定由哪个函数执行操作。函数重载有助于解决复杂问题。

对于函数重载问题,要区分函数名相同的函数,只有从它的参数上进行区分。具体地说,要实现函数重载,它们的参数必须满足以下两个条件之一:

(1) 参数的个数不同。

(2) 参数的类型不同。

6. 简述重载函数的匹配顺序。

答:重载函数选择的规则是按下述顺序将实参类型与被调用的重载函数形参类型一一比较。

(1) 寻找一个严格的匹配,即调用与实参的数据类型、个数完全相同的那个函数。

(2) 通过内部转换寻求一个匹配,即通过(1)的方法没有找到相匹配的函数时,则由C++系统对实参的数据类型进行内部自动转换,转换完毕后,如果有匹配的函数存在,则执行该函数。

(3) 对实参进行强制类型转换,以此作为查找相匹配的函数。

三、编程题

1. 编写一个函数统计任意一个字符串中字母、数字、其他字符出现的频率。

程序如下:

```
#include <iostream.h>
#include <stdio.h>
void count(char letter[ ])
```

```

{   int i,t1,t2,t3,n;
    i=0; t1=0;
    t2=0; t3=0;    n=0;
    while(letter[i]!='\0')
    {
        if((letter[i]>='a'&&letter[i]<='z')|| (letter[i]>='A'&&letter[i]
        <='Z'))
            t1++;
        else if (letter[i]>='0'&& letter[i]<='9')
            t2++;
        else
            t3++;
        i++;
    }
    n=t1+t2+t3;
    cout<<"字母出现的频率为:"<<float(t1)/n<<endl;
    cout<<"数字出现的频率为:"<<float(t2)/n<<endl;
    cout<<"其他字符出现的频率为:"<<float(t3)/n<<endl;
}
void main( )
{
    char str[80];
    gets(str);
    count(str);           //采用传址调用
}

```

2. 设有 10 名学生 3 门考试课, 编写一函数输出平均分不及格的学生。
程序如下:

```

#include <iostream.h>
#include <iomanip.h>
const int n=10;
const int m=3;
int a[n][m];
void average();
void main()
{
    cout<<"请输入 10 个学生的 3 门课程成绩:\n";
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            cin>>a[i][j];
    cout<<endl;
    average( );
}
void average()
{
    for(int i=0;i<n;i++)
    {
        int sum=0;
        for(int j=0;j<m;j++)
            sum+=a[i][j];
    }
}

```

```

        if (double(sum)/m<60)
            cout<<"第"<<i+1<<"个学生平均分不及格"<<endl;
    }
}

```

3. 编程实现用弦截法求方程 $x^3-5x^2+16x-80=0$ 在区间 $[2, 6]$ 内的根。

程序如下:

```

#include <math.h>
float f(float x)
{
    float y;
    y=((x-5.0)*x+16.0)*x-80.0;
    return(y);
}
float xpoint(float x1,float x2)
{
    float y;
    y=(x1*f(x2)-x2*f(x1))/(f(x2)-f(x1));
    return(y);
}
float root(float x1,float x2)
{
    float x,y,y1;
    y1=f(x1);
    do
    {
        x=xpoint(x1,x2);
        y=f(x);
        if(y*y1>0)
        {
            y1=y;
            x1=x;
        }
        else x2=x;
    }while(fabs(y)>=0.0001);
    return(x);
}
#include <iostream.h>
void main()
{
    float x1=2,x2=6,x;
    x=root(x1,x2);
    cout<<"A root of equation is:"<<x<<"\n";
}

```

4. 使用函数重载的方法, 设计两个求面积函数。

程序如下:

```

#include<iostream.h>
const double PI=3.1415;
double area(float r)                //求圆面积,需传递一个参数
{

```