C++程序设计试题及答案解析（二）

C++程序设计模拟试卷(二)

一、单项选择题

1. 静态成员函数没有（）

A. 返回值

B. this指针

C. 指针参数

D. 返回类型

答案：B

解析：(P107)静态成员函数是普通的函数前加入static,它具有函数的所有的特征：返回类型、形参，所以使用(P107)静态成员函数，指针可以作为形参，也具有返回值。静态成员是类具有的属性，不是对象的特征，而this表示的是隐藏的对象的指针，因此静态成员函数没有this 指针。静态成员函数当在类外定义时，要注意不能使用static关键字作为前缀。由于静态成员函数在类中只有一个拷贝（副本），因此它访问对象的成员时要受到一些限制：静态成员函数可以直接访问类中说明的静态成员，但不能直接访问类中说明的非静态成员；若要访问非静态成员时，必须通过参数传递的方式得到相应的对象，再通过对象来访问。

2. 假定AB为一个类，则执行“AB a(2), b［3］,*p［4］;”语句时调用该类构造函数的次数为（）

A. 3

B. 4

C. 5

D. 9

答案：B

解析：(P79)a(2)调用1次带参数的构造函数，b［3］调用3次无参数的构造函数，指针没有给它分配空间，没有调用构造函数。所以共调用构造函数的次数为4。

3. 有关多态性说法不正确的是（）

A. C++语言的多态性分为编译时的多态性和运行时的多态性

B. 编译时的多态性可通过函数重载实现

C. 运行时的多态性可通过模板和虚函数实现

D. 实现运行时多态性的机制称为动态多态性

答案：C

解析：(P171)多态性分为静态的和动态的。静态通过函数的重载来实现，动态是通过基类指针或基类引用和虚函数来实现的。所以错误的是C项。

4. 假定一个类的构造函数为“A(int i=4, int j=0) {a=i;b=j;}”, 则执行“A x (1);”语句后，x.a和x.b的值分别为（）

A. 1和0

B. 1和4

C. 4和0

D. 4和1

答案：A

解析：(P75)带默认的构造函数，对应实参没有值时就采用形参值。调用构造函数时，i=1,不采用默认值，而只有一个参数，j采用默认值0即j=0,因此a=1,b=0,选择A项。

5. 类MyA的拷贝初始化构造函数是 （）

A. MyA()

B. MyA(MyA*)

C. MyA(MyA&)

D. MyA(MyA)

答案：C

解析：(P80)复制即拷贝构造函数使用对象的引用作形参，防止临时产生一个对象，A无参构造函数，B是指针作为形参，D项是对象，所以选择C项。

6. 在C++中，函数原型不能标识（）

A. 函数的返回类型

B. 函数参数的个数

C. 函数参数类型

D. 函数的功能

答案：D

解析：函数的声明，说明函数的参数、返回类型以及函数名，函数体即实现部分决定功能。所以函数的原型不能决定函数的功能。

7. 友元关系不能（）

A. 提高程序的运行效率

B. 是类与类的关系

C. 是一个类的成员函数与另一个类的关系

D. 继承

答案：D

解析：(P111)友元可以是函数与类的关系即友元函数，也可以类与类的关系即友元类，但友元不能继承，是单向性，且不具有传递性。友元可以访问类中所有成员，提高了访问的方便性。因此选择D项。

8. 实现两个相同类型数加法的函数模板的声明是（）

A. add(T x,T y)

B. T add(x,y)

C. T add(T x,y)

D. T add(T x,T y)

答案：D

解析：(P63)实现两个相同类型数加法结果应该和操作数具有相同类型。进行加法运算后结果也是和参数具有相同类型，需要返回值。A无返回值时要用void,B形参无类型，C形参y没有类型，所以选择D项。

9. 在int a=3,int *p=&a；中，*p的值是（）

A. 变量a的地址值

B. 无意义

C. 变量p的地址值

D. 3

答案：D

解析：*p代表引用a变量的值，p代表a的地址值。所以选择D项。

10. 下列不是描述类的成员函数的是（）

A. 构造函数

B. 析构函数

C. 友元函数

D. 拷贝构造函数

答案：C

解析：(P109)构造函数、析构函数、拷贝构造函数都是特殊的成员函数，友元则不是成员函数。所以选择C项。

11. 如果从原有类定义新类可以实现的是（）

A. 信息隐藏

B. 数据封装

C. 继承机制

D. 数据抽象

答案：C

解析：(P129)继承指在原有类的基础上产生新类。数据封装即数据和操作组合在一起，形成类。信息的隐藏，通过访问权限来实现。数据抽象，将事物的特征抽象为数据成员或服务。因此选择C项。

12. 下面有关类说法不正确的是（）

A. 一个类可以有多个构造函数

B. 一个类只有一个析构函数

C. 析构函数需要指定参数

D. 在一个类中可以说明具有类类型的数据成员

答案：C

解析：(P80)构造函数可以有参数、可以重载、因此可以有多个，A项正确。析构函数只有一个不能重载、不能继承，没有返回值，B项正确，C项错误。

13. 在函数定义中的形参属于（）

A. 全局变量

B. 局部变量

C. 静态变量

D. 寄存器变量

答案：B

解析：形参或函数中定义的变量都是局部变量。在函数外定义的变量是全局变量。形参只能用局部变量，频繁使用的变量可以声明为寄存器变量，形参不能使用静态变量或寄存器变量。

14. 下列有关重载函数的说法中正确的是（）

A. 重载函数必须具有不同的返回值类型

B. 重载函数参数个数必须相同

C. 重载函数必须有不同的形参列表

D. 重载函数名可以不同

答案：C

解析：(P59)函数的重载必须函数名相同而形参类型或个数不同，与返回值无关。

15. this指针存在的目的是（）

A. 保证基类私有成员在子类中可以被访问

B. 保证基类保护成员在子类中可以被访问

C. 保证每个对象拥有自己的数据成员，但共享处理这些数据成员的代码

D. 保证基类公有成员在子类中可以被访问

答案：C

解析：(P86)C++要求函数在被调用之前，应当让编译器知道该函数的原型，以便编译器利用函数原型提供的信息去检查调用的合法性，强制参数转换成为适当类型，保证参数的正确传递。对于

标准库函数，其声明在头文件中，可以用#include宏命令包含这些原型文件；对于用户自定义函数，先定义、后调用的函数可以不用声明，但后定义、先调用的函数必须声明。一般为增加程序的可理解性，常将主函数放在程序开头，这样需要在主函数前对其所调用的函数一一进行声明，以消除函数所在位置的影响。所以选择C项。

16. 关于new运算符的下列描述中，错误的是（）

A. 它可以用来动态创建对象和对象数组

B. 使用它创建的对象或对象数组可以使用运算符delete删除

C. 使用它创建对象时要调用构造函数

D. 使用它创建对象数组时必须指定初始值

答案：D

解析：(P78)new创建的对象数组不能指定初始值，所以调用无参的构造函数，选择D项。

17. 已知：p是一个指向类A数据成员m的指针，A1是类A的一个对象。如果要给m赋值为5，正确的是（）

A. A1.p=5;

B. A1->p=5;

C. A1.*p=5;

D. *A1.p=5;

答案：C

解析：(P118)A中p是指针即地址，错误；B选项中A1不是指针不能使用指向运算符->,错误；“*”比“.”级别要高，所以D选项*A1.p=5相当于(*A1).p=5;错误。另外涉及到指向成员函数时注意以下几点：

指向成员函数的指针必须于其赋值的函数类型匹配的三个方面：(1)参数类型和个数；(2)返回类型；(3)它所属的类类型。

成员函数指针的声明：指向short型的Screen类的成员的指针定义如下：

short Screen::* ps_Screen;

ps_Screen可以用_height的地址初始化如下：short Screen::*ps_Screen=&Screen::_height;

类成员的指针必须总是通过特定的对象或指向改类型的对象的指针来访问。是通过使用两个指向成员操作符的指针(针对类对象和引用的.*，以及针对指向类对象的指针的->*)。

18. 以下基类中的成员函数表示纯虚函数的是（）

A. virtual void tt()=0

B. void tt(int)=0

C. virtual void tt(int)

D. virtual void tt(int){}

答案：A

解析：(P173)当在基类中不能为虚函数给出一个有意义的实现时，可以将其声明为纯虚函数，实现由派生类完成。格式：virtual<函数返回类型说明符><函数名>(<参数表>)=0;。

19. C++类体系中，不能被派生类继承的有（）

A. 常成员函数

B. 构造函数

C. 虚函数

D. 静态成员函数

答案：B

解析：(P132)构造函数不能被继承。

20. 静态成员函数不能说明为（）

A. 整型函数

B. 浮点函数

C. 虚函数

D. 字符型函数

答案：C

解析：(P108)使用关键字static声明的成员函数就是静态成员函数，静态成员函数也属于整个类而不属于类中的某个对象，它是该类的所有对象共享的成员函数。

静态成员函数可以在类体内定义，也可以在类外定义。当在类外定义时，要注意不能使用static关键字作为前缀。

由于静态成员函数在类中只有一个拷贝（副本），因此它访问对象的成员时要受到一些限制：静态成员函数可以直接访问类中说明的静态成员，但不能直接访问类中说明的非静态成员；若要访问非静态成员时，必须通过参数传递的方式得到相应的对象，再通过对象来访问。虚函数是非静态的、非内联的成员函数。静态成员函数不能被说明为虚函数。

二、填空题(本大题共20小题，每小题1分，共20分)请在每小题的空格中填上正确答案。错填、不填均无分。

1. 假设int a=1,b=2;则表达式(++a/b)*b--的值为___。

答案：2

［解析］前缀＋＋或——表示先使变量值变化，再使用，这和后缀恰恰相反。但是编译cout<<(++a/b)*b--时，先++a/b值为1，后1*b--，先取b＝2，结果为2，再让b＝1。

2. 抽象类中至少要有一个___函数。

答案：(P173)纯虚

［解析］至少有一个纯虚函数的类就称为抽象类，即不能实例化。

3. 一个抽象类的派生类可以实例化的必要条件是实现了所有的___。

答案：(P173)纯虚函数的定义

［解析］抽象类只因有纯虚函数，所以不能被实例化，所以派生类要实例化必须对纯虚函数进行定义。

4. 下面程序的输出结果为___。

```
#include <iostream.h>
void main()
{int num=2,i=6;
do
{i--;
num++;
}while(--i);
cout<<num<<endl;
}
```

答案：5

［解析］do - while循环，前缀先使i减少1后判断是否为零，不为零时再次执行循环，为零退出循环。循环值执行3次就退出，所以结果为5。

5. 静态成员函数、友元函数、构造函数和析构函数中，不属于成员函数的是___。

答案：(P109)友元函数

［解析］友元函数不是类成员，但可以访问类成员。类的封装性保证了数据的安全，但引入友元，虽然访问类是方便了，但确实破坏类访问的安全性。

6. 在用C＋＋进行程序设计时，最好用___代替malloc。

答案：(P10)new

［解析］new与delete是C++语言特有的运算符，用于动态分配和释放内存。new用于为各种数据类型分配内存，并把分配到的内存首地址赋给相应的指针。new的功能类似于malloc（）函数。

使用new的格式为：

<指针变量>new<数据类型>;

其中，<数据类型>可以是基本数据类型，也可以是由基本类型派生出来的类型；<指针变量>取得分配到的内存首地址。new有3种使用形式。

（1）给单个对象申请分配内存

int *ip;ip=new int;//ip指向1个未初始化的int型对象

（2）给单个对象申请分配内存的同时初始化该对象

int *ip;ip=new int(68);//ip指向1个表示为68的int型对象

（3）同时给多个对象申请分配内存

int *ip;ip=new int［5］;//ip指向5个未初始化的int型对象的首地址

for(int i=0;i<5;i++)ip［i］=5*i+1;//给ip指向的5个对象赋值

用new申请分配内存时，不一定能申请成功。若申请失败，则返回NULL，即空指针。因此，在程序中可以通过判断new的返回值是否为0来获知系统中是否有足够的空间供用户使用。

7. 由const修饰的对象称为___。

答案：(P113)常对象

［解析］使用const关键字说明的成员函数称为常成员函数，使用const关键字说明的对象称为常对象。

常成员函数的说明格式如下：<返回类型说明符><成员函数名>(<参数表>)const;

常成员函数不更新对象的数据成员，也不能调用该类中没有用const修饰的成员函数。常对象只能调用它的常成员函数，而不能调用其他成员函数。const关键字可以用于参与重载函数的区分。

8. 在C++程序设计中，建立继承关系倒挂的树应使用___继承。

答案：(P138)单

［解析］一个基类可以派生多个子类，一个子类可以再派生出多个子类，这样就形成了一个倒立的树。

9. 基类的公有成员在派生类中的访问权限由___决定。

答案：(P132)访问控制方式或继承方式

10. 不同对象可以调用相同名称的函数，但执行完全不同行为的现象称为___。

答案：(P167)多态性

［解析］多态性的概念。虚函数是实现多态的基础，运行过程中的多态需要同时满足3个条件：(1)类之间应满足子类型关系。(2)必须要有声明的虚函数。(3)调用虚函数操作的是指向对象的指针或者对象引用；或者是由成员函数调用虚函数（如果是在构造函数或析构函数中调用虚函数，则采用静态联编）。

11. this指针始终指向调用成员函数的___。

答案：对象

this指针是隐藏的指针，它指向调用函数的对象。

12. 预处理命令以___符号开头。

答案：(P183)operater

［解析］文件包含、预处理和编译都是以＃开头。

13. 类模板用来表达具有___的模板类对象集。

答案：(P145)相同处理方法

［解析］模板特点是不同的数据具有相同的处理方法的抽象。

14. C++程序的源文件扩展名为___。

答案：(P21)cpp

［解析］源程序*.cpp,目标文件为*.obj,可执行程序*.exe。

15. 在#include命令中所包含的头文件，可以是系统定义的头文件，也可以是___的头文件。

答案：(P7)自定义

［解析］＃include装入文件有两种方式<>和“”，一是系统的，一是自定义文件。

16. vector类中向向量尾部插入一个对象的方法是___。

答案：(P157)push_back

17. C++语言中如果调用函数时，需要改变实参或者返回多个值，应该采取___方式。

答案：(P51)传地址或引用

［解析］传地址即指针，在函数中通过指针修改它指向的变量的值时，实参也就变化了。使用引用，直接修改变量的别名即引用的值，该变量也就随着变化。

18. 语句序列

ifstream infile;

infile.open(“data.dat”);

的功能可用一个语句实现，这个语句是___。

答案：(P199)ifstream infile(“data.dat”);

［解析］void ifstream::open(const char *fname,int mode=ios::in,int access=filebuf::openprot);

ifstream::ifstream(const char *fname,int mode=ios::in,int access=filebuf::openprot);

其中，第一个参数是用来传递文件名的；第二个参数mode的值决定文件将如何被打开；第三个参数access的值决定文件的访问方式，一般取缺省值filebuf::openprot，表示是普通文件。

mode的取值如下：(1)ios::in：打开一个文件进行读操作，而且该文件必须已经存在；(2)ios::nocreate：不建立新的文件。当文件不存在时，导致open()失败；(3)ios::noreplace：不修改原来已经存在的文件。若文件已经存在，导致open()失败；(4)ios::binary：文件以二进制方式打开，缺省时为文本文件。

19. 如果要把类B的成员函数void fun()说明为类A的友元函数，则应在类A中加入语句___。

答案：(P111)friend void B::fun();

［解析］声明成员函数作为另外一个类的友元函数时，使用类作用域运算符：：。

20. 在编译指令中，宏定义使用___指令。

答案：(P6、97)#define

［解析］静态成员是所有对象共享的特征，也就是类的特征。

三、改错题(本大题共5小题，每小题2分，共10分)下面的类定义中有一处错误，请用下横线标出错误所在行并给出修改意见。

1. #include <iostream>

#include <fstream>

#include <string>

using namespace std;

class A

{public:

A(const char *na){strcpy(name,na);}

private:

char name［80］;

```
};
class B:public A
{ public:
B(const char *nm):A(nm){}
void show();
};
void B::show()
{ cout<<"name:"<<name<<endl;
}
void main()
{ B b1("B");
b1.show();
}
```

答案：private:因为name如果是私有的，在派生类中无法访问，而基类没有提供成员函数来访问name，所以更改name访问权限为公有或保护，这样对于派生类来说是透明的。

［修改］public：或protected：

2.

```
#include <iostream.h>
void f(int *a,int n)
{int i=0,j=0;
int k=0;
for(;i<n/2;i++)
{k=a［i］;
a［i］=a［n-i-1］;
a［n-i-1］=k;
}
}
void show(int a［］,int n)
{for(int i=0;i<n;i++)
cout<<a［i］<<" ";
cout<<endl;
}
void main()
{int p［5］;
int i=0,n=5;
for(;i<5;i++)
{p［i］=i;}
f(*p,n);
show(p,n);
```

答案：　［修改］f(p,n);

［解析］f(*p,n);f函数第一个参数是指针而调用时使用*p，*p表示p所指向的变量或对象，不是地址即不是指针。

3.

```
#include <iostream.h>
void main()
{int i(3),j(8);
```

```
int * const p=&i;
cout<<*p<<endl;
p=&j;
cout<<*p<<endl;
}
```

答案：int * const p=&i;在指针变量前加const表示一个常指针即地址不能变化，它指向的变量不能改变且定义时必须设置指向变量或对象的地址。

［修改］int *p=&i;

4.

```
#include <iostream.h>
void main()
{int i,*p;
i=10;
*p=i;
cout<<*p<<endl;
}
```

答案：*p=i;指针即地址没有被赋值。

［修改］p=&i;

5.

```
#include <iostream.h>
class A
{private:
int x,y;
public:
void fun(int i,int j)
{x=i;y=j;}
void show()
{cout<<x<<" "<<y<<endl;}
};
void main()
{A a1;
a1.fun(2);
a1.show();
}
```

答案：void fun(int i,int j)调用时有一个参数，形参有两个，可以使第二个带默认值。

［修改］void fun(int i,int j = 0)

四、完成程序题(本大题共5小题，每小题4分，共20分)

1. 完成下面类中成员函数的定义。

```
#include <iostream>
#include <string>
using namespace std;
class str
{private:
char *st;
public:
str(char *a)
```

{set(a);

}

str & operator=(____)

{delete st;

set(a.st);

return *this;

}

void show(){cout<<st<<endl;}

~str(){delete st;}

void set(char *s)//初始化st

{_____

strcpy(st,s);

}

};

void main()

{str s1("he"),s2("she");

s1.show(),s2.show();

s2=s1;

s1.show(),s2.show();}

答案：str &a，st=new char［strlen(s)+1］；

［解析］对“＝”运算符进行重载，调用时s2=s1,都是对象，所以形参使用对象的引用，不要使用对象作为形参（产生临时对象）。使用strcpy进行字符的复制，st必须有一定的空间，空间是strlen(s)+1（‘\0’作为结束符，strlen得到的长度不包括结束符）。

2. 一个类的头文件如下所示，num初始化值为5，程序产生对象T，且修改num为10，并使用show()函数输出num的值10。

#include <iostream.h>

class Test

{private:

static int num;

public:

Test(int);

void show();

};

_____

Test::Test(int n)

{num=n;}

void Test::show()

{cout<<num<<endl;}

void main()

{Test t(10);

_____

}

答案：int Test::num=5;，t.show();

［解析］静态成员在类外初始化，注意它的格式。调用show输出。

3. 下面是一个三角形三边，输出其面积C++程序，在下划线处填上正确的语句。

```
#include <iostream.h>
#include <math.h>
void area()
{double a,b,c;
cout<<"Input a b c:";
_________
if(a+b>c&&a+c>b&&c+b>a)
{double l=(a+b+c)/2;
_______
cout<<"The area is:"<<s<<endl;
}
else
cout<<"Error"<<endl;
}
void main()
{area();}
```

答案：cin>>a>>b>>c;，double s=sqrt(l*(l-a)*(l-b)*(l-c));

［解析］输入三个边的长度，由公式得出三角形的面积double s=sqrt(l*(l-a)*(l-b)*(l-c));

4. 下面程序中Base是抽象类。请在下面程序的横线处填上适当内容，以使程序完整,并使程序的输出为:

```
Der1 called!
Der2 called!
#include <iostream.h>
class Base
{public:
_________________;
};
class Der1:public Base
{public:
void display(){cout<<"Der1 called!"<<endl;}
};
class Der2:public Base
{public:
void display(){cout<<"Der2 called!"<<endl;}
};
void fun(_______________)
{p->display();}
void main()
{Der1 b1;
Der2 b2;
Base * p=&b1;
fun(p);
p=&b2;
```

fun(p);

}

答案：virtual void display()=0;，Base *p

［解析］抽象类有纯虚函数，派生类为display。结果fun函数用指针做参数。

5. 下面程序中用来求数组和。请在下面程序的横线处填上适当内容，以使程序完整,并使程序的输出为:s = 150。

#include <iostream.h>

class Arr

{int *a,n;

public:

Arr():a(0),n(0){}

Arr(int *aa, int nn)

{n=nn;

a=new int［n］;

for(int i=0;i<nn;i++)

(a+i)=(aa+i);

}

~Arr(){delete a;}

_____;

{return *(a+i);}

};

void main()

{int b［5］={10,20,30,40,50};

Arr a1(b,5);

int i=0,s=0;

_____

s+=a1.GetValue(i);

cout<<"s="<<s<<endl;

}

答案：int GetValue(int i)，for(;i<5;i++)

［解析］函数调用GetValue,由此可知要定义该函数，循环求和，循环5次。

五、程序分析题(本大题共4小题，每小题5分，共20分)

1. 给出下面程序输出结果。

#include <iostream.h>

class example

{int a;

public:

example(int b=5){a=b++;}

void print(){a=a+1;cout <<a<<"";}

void print()const

{cout<<a<<endl;}

};

void main()

{example x;

```
const example y(2);
x.print();
y.print();
}
```

答案：62

［解析］x是普通对象，调用普通的print函数；而y常对象，调用常成员函数。

2. 给出下面程序输出结果。

```
#include <iostream.h>
void main()
{ int *p1;
int **p2=&p1;
int b=20;
p1=&b;
cout<<**p2<<endl;
}
```

答案：20

［解析］p1指向b，而p指向p1的地址。*p2表示p1的地址，p1的地址就是&b，即*p2是&b,所以**p2就是b变量的值。

3. 给出下面程序输出结果。

```
#include <iostream.h>
class Base
{private:
int Y;
public:
Base(int y=0) {Y=y;cout<<"Base("<<y<<") \ n";}
~Base() {cout<<"~Base() \ n";}
void print() {cout <<Y<< "";}
};
class Derived:public Base
{private:
int Z;
public:
Derived (int y, int z):Base(y)
{Z=z;
cout<<"Derived("<<y<<","<<z<<") \ n";
}
~Derived() {cout<<" ~ Derived() \ n";}
void print()
{Base::print();
cout<<Z<<endl;
}
};
void main()
{Derived d(10,20);
```

d.print();

}

答案：Base(10)

Derived(10,20)

10 20

～Derived()

~Base()

［解析］派生类对象，先调用基类构造函数输出Base(10)，后调用派生类构造函数输出

Derived(10,20)，后执行d.print(),调用派生类的print，再调用Base::print()输出10，后返回

输出z的值20。后派生类析构，再基类析构。

4. 给出下面程序输出结果。

#include <iostream.h>

class A

{public:

A()

{cout<<"A 构造函数 \ n";fun();}

virtual void fun()

{cout<<"A::fun() 函数 \ n";}

};

class B:public A

{public:

B()

{cout<<"B构造函数 \ n";fun();}

void fun() {cout<<"B::fun() calle函数 \ n";}

};

void main()

{B d;}

答案：A构造函数

A::fun()函数

B构造函数

B::fun()calle函数

［解析］定义派生类对象，首先调用基类构造函数，调用A类中fun()，然后调用B类的构造函数

，在调用B的fun函数。

六、程序设计题(本大题共1小题，共10分)

1. 编写类String的构造函数、析构函数和赋值函数和测试程序。

已知类String的原型为：

#include <iostream.h>

#include <string.h>

class String

{public:

String(const char *str=NULL); // 普通构造函数

String(const String &other); // 拷贝构造函数

~String(); // 析构函数

String & operator=(const String &other); // 赋值函数

```
void show()
{cout<<m_data<<endl;
}
private:
char *m_data; // 用于保存字符串
};
```

答案：String::~String()

```
{delete［］m_data;//由于m_data是内部数据类型，也可以写成delete m_data;
}
String::String(const char *str)
{if(str==NULL)
{m_data=new char［1］;//若能加NULL判断则更好
*m_data= \ 0;
}
else
{int length=strlen(str);
m_data=new char［length+1］; //若能加NULL判断则更好
strcpy(m_data, str);
}
}
String::String(const String &other)
{int length=strlen(other.m_data);
m_data=new char［length+1］;//若能加NULL判断则更好
strcpy(m_data, other.m_data);
}
String & String::operator=(const String &other)
{if(this==&other)
return *this;
delete［］m_data;
int length=strlen(other.m_data);
m_data=new char［length+1］;//若能加NULL判断则更好
strcpy(m_data, other.m_data);
return *this;
}
void main()
{String str1("aa"),str2;
str1.show();
str2=str1;
str2.show();
String str3(str2);
str3.show();
}
```

C++程序设计试题及答案解析（三）