

一、选择题

1 在 E—R 图中，用来表示实体联系的图形是()。

A. 椭圆形 B. 矩形 C. 菱形 D. 三角形

参考答案：C

参考解析：在 E—R 图中实体集用矩形，属性用椭圆，联系用菱形。

2 有如下程序：

```
#include<iostream>

using namespace std; class Part{
public:
    Part(int x=0): val(x){cout<< "Part: " << x << endl;}
private:
    int val;
};

class Whole{ public:
    Whole(int x, int Y, int z=0): p2(x), p1(y), val(z){cout<< "Whole: " << x << " " << Y << " " << z << endl;}
private:
    Part p1, p2; int val;
};

int main() {
    Whole obj(1, 2, 3); return 0;
}
```

执行这个程序的输出结果是()。

A. 123321B. 213312C. 213D. 123123

参考答案: B

参考解析: 本题考查带参数的构造函数。本题中定义了一个对象 obj, 那么执行其构造函数, 初始化 p1, p2 的顺序是由声明这两个成员的顺序决定的, 而不是由参数化列表顺序决定, 所以本题中, 会先执行 p1, 输出 2, 然后执行 p2 输出 1, 然后输出 3, 最后一次输出 312。所以本题答案为 8。

3 没有定义 `int x; float Y;`, 则 `10+x+y` 值的数据类型是()。

A. intB. doubleC. floatD. 不确定

参考答案: C

参考解析: 由算术运算符组成的表达式中, 若含有不同类型的操作数, 则系统隐含地将低类型转化为高类型, 由低到高的顺序为 `int→float→double`。

4 有如下类定义:

```
class Point{  
private:  
static int how_many;
```

```
};  
_____  
how_many=0;
```

要初始化 Point 类的静态成员 how_many, 下画线处应填入的内容是()。

A. int B. static int C. int Point: : D. static int Point: :

参考答案: C

参考解析: 本题考查静态数据成员, 静态数据成员可以初始化, 但只能在类体外进行初始化, 其一般形式为: 数据类型类名: : 静态数据成员名=初值。所以本题答案为 C。

5 有如下类定义：

```
class Test  
{  
  
public:  
  
Test() {a=0;c=0;}//①  
  
int f(im A.const{this->a=a;}//②  
  
static int g(){return a;f//③  
  
void h(int B. {Test: .b=b;};//④  
  
private:  
  
int a;  
  
static int b;  
  
const int C;  
  
};  
  
int test : b=0;
```

在标注号码的行中，能被正确编译的是()。

A. ①B. ②C. ③D. ④

参考答案：D

答案解析：只能通过构造函数的参数初始化列表对常数据成员进行初始化，本题中常数据成员为C。①通过默认构造函数初始化c，所以不正确。常成员函数只能引用本类中数据成员，而不能修改它，所以②不正确。静态成员函数由于没有this指针，所以不能访问本类中的非静态成员，所以

③错误。

6 某系统总体结构图如下图所示：



该系统总体结构图的深度是()。

A. 7B. 6C. 3D. 2

参考答案：C

参考解析：根据总体结构图可以看出该树的深度为3，比如：XY系统——功能2——功能2.1，就是最深的度数的一个表现。

7 有如下类定义：

```
Class AA{ int a; public: AA(int n=0): a(n) {}  
};
```

```
Class BB: public AA{ public:
```

```
BB(int n)——};
```

其中横线处缺失的部分是()。

A. : a(n) {} B. : AA(n) {} C. {a(n);} D. {a=n;}

参考答案：B

参考解析：本题考查公用继承对基类成员访问属性。公用继承中，基类的公用成员和保护成员在派生类中保持原有访问属性，其私有成员仍为基类私有，本题中若要给派生类继承下基类的私有成员赋值，那么只能调用基类的成员函数来对它赋值，所以本题答案为B。

8 有如下类定义：

```
class MyClass{
```

```
int value;

public:

MyClass(int n): value(n) {}

int getValue()const{return value;}

};
```

则类 MyClass 的构造函数的个数是()。

A. 1 个 B. 2 个 C. 3 个 D. 4 个

参考答案: B

参考解析: 本题考查类的默认构造函数和自定义的构造函数, 本题中 MyClass(int n): value(n) {} 为自定义的构造函数, 加上默认构造函数, 一共有 2 个构造函数。

9 下列关于派生类构造函数和析构函数的说法中, 错误的是()。

- A. 派生类的构造函数会隐含调用基类的构造函数
- B. 如果基类中没有定义构造函数, 那么派生类必须定义构造函数
- C. 在建立派生类对象时, 先调用基类的构造函数, 再调用派生类的构造函数
- D. 在销毁派生类对象时, 先调用基类的析构函数, 再调用派生类的析构函数

参考答案: D

参考解析: 在销毁派生类对象时, 先调用派生类的析构函数, 再调用基类的析构函数。

10 下列叙述中正确的是()。

- A. 栈是一种先进先出的线性表
- B. 队列是一种后进先出的线性表
- C. 栈与队列都是非线性结构

D. 以上三种说法都不对

参考答案: D

参考解析: 栈是一种先进后出的线性表, 队列是一种先进先出的线性表, 栈与队列都是线性结构。

11 关于关键字 `class` 和 `typename`, 下列表述中正确的是()。

A. 程序中的 `typename` 都可以替换为 `class`

B. 程序中的 `class` 都可以替换为 `typename`

C. 在模板形参表中只能用 `typename` 来声明参数的类型

D. 在模板形参表中只能用 `class` 或 `typename` 来声明参数的类型

参考答案: A

参考解析: 程序中的 `typename` 都可以替换为 `class`, 但程序中的 `class` 不可以全部替换为 `typename`。在模板类型形参中除了 `typename` 和 `class` 以外, 也可以用常规类型来声明参数的类型, 所以 C、D 选项错误。

12 数据库管理系统是()。

A. 操作系统的一部分

B. 在操作系统支持下的系统软件

C. 一种编译系统

D. 一种操作系统

参考答案: B

参考解析: 数据库管理系统是数据库的机构, 它是一种系统软件, 负责数据库中数据组织、数据操纵、数据维护、控制及保护和数据服务等。是一种在操作系统之上的系统软件。

13 软件生命周期是指()。

- A. 软件产品从提出、实现、使用维护到停止使用退役的过程
- B. 软件从需求分析、设计、实现到测试完成的过程
- C. 软件的开发过程
- D. 软件的运行维护过程

参考答案: A

参考解析: 通常, 将软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期。也就是说, 软件产品从考虑其概念开始, 到该软件产品不能使用为止的整个时期都属于软件生命周期。

14 下列符号中不属于 C++ 关键字的是()。

- A. friend B. namespace C. Continue D. byte

参考答案: D

参考解析: 本题考查 C++ 关键字, 属于基本常识, 显然 D 选项不正确。

15 下面关于数组的初始化正确的是()。

- A. char str[]={'a', 'b', 'c'}
- B. char str[2]={'a', 'b', 'c'}
- C. char str[2][3]={{'a', 'b'}, {'c', 'd'}, {'e', 'f'}}
- D. char str()={~a, ~b, ~c}

参考答案: A

参考解析: 本题考查的是字符数组的初始化。选项 B 中用 3 个元素初始化大小为 2 的数组, 越界了; 选项 c 中应该是 2 行 3 列的数组, 题中使用 3 行 2 列初始化; 选项 D 中数组组合后应用方括号。

16 下列叙述中正确的是()。

- A. 顺序存储结构的存储一定是连续的, 链式存储结构的存储空间不一定是连续的

- B. 顺序存储结构只针对线性结构，链式存储结构只针对非线性结构
- C. 顺序存储结构能存储有序表，链式存储结构不能存储有序表
- D. 链式存储结构比顺序存储结构节省存储空间

参考答案：A

参考解析：链式存储结构既可以针对线性结构也可以针对非线性结构，所以 B 与 C 错误。链式存储结构中每个结点都由数据域与指针域两部分组成，增加了存储空间，所以 D 错误。

17 下列各组类型声明符中，含义相同的一组是()。

- A. unsigned long int 和 long
- B. signed short int 和 short
- C. unsigned short 和 short
- D. short int 和 int

参考答案：B

参考解析：本题考查整型变量的类型，A、c 选项中前者为无符号的，后者为有符号的，D 选项 short int 为 2 个字节，int 为 4 个字节。

18 下面程序的运行结果是()。

```
#include <iostream>
using namespace std;
class base{
protected: int a; public:
base() {cout<<"0" ;}
}; class base1: virtual base{
public:
base1() {cout<<"1" ;}
```



```

};

class base2: virtual base1 public:
    base2() {cout<<" 2" ;}

};

class derived: public base1, public base2{ public:

    derived() {cout<<" 3" ;}

};

void main()

    derived obj;

    cout<

```

A. 0123B. 3120C. 0312D. 3012

参考答案: A

参考解析: 本题考查的是含有虚基类的继承中构造函数的调用顺序, 应该先调用基类的构造函数, 接着是按照派生类继承列表的顺序依次调用虚基类的构造函数, 最后调用派生类自己的构造函数。

19. 下列描述正确的是()。

A. 表示 $m > n$ 为 true 或 $m \&\& m$

参考答案: D

参考解析: 本题主要考查 c++ 语言的基本语法, 选项 A 中应该为 $m > n \mid j \mid m$

20. 下列运算符函数中肯定不属于类 FunNumber 的成员函数的是()。

A. `int operator -(FunNumber);`

B. `FunNumber operator -();`

C. `FunNumber operator -(int);`

D. int operator -(FunNumber, FunNumber);

参考答案: D

参考解析: 本题考查运算符重载的一些规则: ①一般情况下, 单目运算符最好重载为类的成员函数, 双目运算符则最好重载为类的友元函数; ②双目运算符=、()、[]、->不能重载为类的友元函数; ③类型转换函数只能定义为一个类的成员函数, 而不能定义为类的友元函数; ④若一个运算符的操作需要修改对象的状态, 选择重载为成员函数较好; ⑤若运算符所需的操作数(尤其是第一个操作数)希望有隐式类型转换, 则只能选用友元函数; ⑥当运算符函数是一个成员函数时, 最左边的操作数(或者只有最左边的操作数)必须是运算符类的一个类对象(或者是对该类对象的引用); 如果左边的操作数必须是一个不同类的对象, 或者是一个内部类型的对象, 该运算符函数必须作为一个友元函数来实现。所以答案为 D。

21 下列程序的运行结果是()。

```
#include
```

```
class A
```

```
{
```

```
int a;
```

```
public
```

```
A() {a=0;}
```

```
A(int aa)
```

```
{
```

```
a=aa;
```

```
cout<
```

```
}
```

```
};
```

```
void main()

{

A X, Y(2), z(3);

cout<

}
```

A. 00B. 23C. 34D. 25

参考答案: B

参考解析: 本题考查的是对构造函数的掌握, 另外“++”运算符是右结合的, 所以在进行输出的时候都是先把原来的输出后再自加 1。

22 关于函数重载, 下列叙述中错误的是()。

- A. 重载函数的函数名必须相同
- B. 重载函数必须在参数个数或类型上有所不同
- C. 重载函数的返回值类型必须相同
- D. 重载函数的函数体可以有所不同

参考答案: C

参考解析: 所谓函数重载是指同一个函数名可以对应多个函数的实现。每种实现对应一个函数体, 这些函数的名字相同, 其参数类型或个数不同, 但不能出现函数返回值类型不同而参数类型和个数完全相同的重载。所以本题答案为 c。

23 在 c++ 语言中, 数据封装要解决的问题是()。

- A. 数据的规范化
- B. 便于数据转换
- C. 避免数据丢失

D. 防止不同模块之间数据的非法访问

参考答案: D

参考解析: 数据封装为了防止不同模块间数据的非法访问及修改。

24 在 C++ 中, cin 是一个()。

A. 类 B. 对象 C. 模板 D. 函数

参考答案: B

参考解析: 本题考查流对象 cin, 属于基础知识, cin 是输入流对象, 负责从键盘获取数据, 然后送到内存。本题答案为 B。

25 在公有派生情况下, 有关派生类对象和基类对象的关系, 下列叙述不正确的是()。

- A. 派生类的对象可以赋给基类的对象
- B. 派生类的对象可以初始化基类的引用
- C. 派生类的对象可以直接访问基类中的成员
- D. 派生类的对象的地址可以赋给指向基类的指针

参考答案: C

参考解析: 公有继承的派生类和基类是子类型的关系, 所谓子类型是类型间一般和特殊的关系, 即派生类是基类的子类型或者说基类的操作可以被用于操作派生类的对象。

26 要建立文件流并打开当前目录下的文件 file.dat 用于输入, 下列语句中错误的是()。

- A. ifstream fin=fstream.open(" file.dat");
- B. ifstream*fin=new ifstream(" file.dat");
- C. ifstream fin;fin.open(" file.dat");

D. ifstream*fin=new ifstream();fin->open(" file.dat");

参考答案: A

参考解析: 本题考查对文件流输入的操作, 可以调用文件流的成员函数 open, 一般形式如下: 文件流对象.open(磁盘文件名, 输入/输出方式)。根据以上格式, 可以判断 A 选项错误。

27 某二叉树有 5 个度为 2 的结点, 则该二叉树中的叶子结点数是()。

A. 10 B. 8 C. 6 D. 4

参考答案: C

参考解析: 根据二叉树的基本性质 3: 在任意一颗二叉树中, 度为 0 的叶子结点总是比度为 2 的结点多一个, 所以本题中叶子结点数=5+1=6 个。

28 下面的语句中错误的是()。

A. int a=5;int x[a]; B. const int a=5;int x[a]; C. int n=5;int *p=new int[a]; D. const int n=5;int *p=new int[a];

参考答案: A

参考解析: 在 C++ 中定义数组时, 数组的范围参数在静态定义时必须是一个常数, 而在动态分配内存定义时可以是常数, 也可以是一个赋了值的参数故选项 A 是错误的定义方法。

29 if 语句的语法格式可描述为:

格式 1: if(<条件>)<语句>

或

格式 2: if(<条件>)<语句 1>

else<语句 2>

关于上面的语法格式, 下列表述中错误的是()。

A. <条件>部分可以是一个 if 语句, 例如 if(if(a==0)...)...

B. <语句>部分可以是一个 if 语句，例如 if(...)if(...)...

C. 如果在<条件>前加上逻辑非运算符!并交换<语句 1>和<语句 2>的位置，语句功能不变

D. <语句>部分可以是一个循环语句，例如 if(...)while(...)...

参考答案：A

参考解析：本题考查 if 语句，if 语句的条件部分不能嵌套 if 语句，所以 A 选项错误；B 选项就是 if 语句嵌套；C 选项其实就是将判断条件颠倒，然后执行语句也颠倒，所以效果一样；D 选项也正确。

30 结构化程序设计的基本原则不包括()。

A. 多态性 B. 自顶向下 C. 模块化 D. 逐步求精

参考答案：A

参考解析：结构化程序设计的思想包括：自顶向下、逐步求精、模块化、限制使用 goto 语句，所以选择 A。

31 下列有关继承和派生的叙述中，正确的是()。

A. 派生类不能访问基类的保护成员

B. 作为虚基类的类不能被实例化

C. 派生类应当向基类的构造函数传递参数

D. 虚函数必须在派生类中重新实现

参考答案：

参考解析：派生类可以访问基类的保护成员，而不能访问基类的私有成员。作为虚基类的类可以被实例化。虚函数如果没有派生类中重新实现，那么仍然使用基类的成员函数。

32 下列关于 this 指针的描述中，正确的是()。

A. 类的成员函数都有 this 指针

- B. 类的友元函数都有 this 指针
- C. 任何与类相关的函数都有 this 指针
- D. 类的非静态成员函数都有 this 指针

参考答案: D

参考解析: this 作用域是在类内部, 当在类的非静态成员函数中访问类的非静态成员时, 编译器会自动将对象本身的地址作为一个隐含参数传递给函数。另外, 全局函数和静态函数都不能使用 this 指针。所以答案为 D。

33 已知函数 f 的原型是 void f(int*a, long&b.;, 变量 v1、v2 的定义是:

```
int v1; long v2;
```

下列调用语句中正确的是()。

A. f(v1, &v2); B. f(v1, v2); C. f(&v1, &v2); D. f(&v1, v2);

参考答案: D

参考解析: 函数参数的 3 种传递方式: ①将变量名作为形参和实参, 即传值方式; ②传递变量指针; ③引用形参, 本题中函数 f 的第一个形参为指针, 那么调用 f 函数时, 第一个参数必须是地址, 所以必须是 &v1. 第二个形参为引用, 所以对应实参为 v2 即可。故本题答案为 D。

34 设有表示学生选课的五张表, 学生 s(学号, 姓名, 性别, 年龄, 身份证号), 课程 c(课号, 课名), 选课 Sc(学号, 课号, 成绩), 则表 Sc 的关键字(键或码)为()。

- A. 课号, 成绩
- B. 学号, 成绩
- C. 学号, 课号
- D. 学号, 姓名, 成绩

参考答案: C

参考解析：学号是学生表 s 的主键，课号是课程表 c 的主键，所以选课表 sc 的关键字就应该是与前两个表能够直接联系且能唯一定义的学号和课号，所以选择 c。

35 有如下程序：

```
#include using namespace 耐: Class MyClass{

public:

MyClass() {++Count;}

~MyClass() {--Count;}

static int getCount() {return Count;} private:

static int Count;

}; ‘ int MyClass: : Count=0;

int main() {

MyClass obj;

cout<<

MyClass+ptr=new MyClass: cout<<

cout<<

}
```

执行这个程序的输出结果是()。

A. 121B. 252C. 221D. 122

参考答案：A

参考解析：本题考查静态数据成员。静态数据成员在内存中只占一份空间，静态数据成员属于类，即使不定义对象，也会给静态数据成员分配空间，可以被引用。本题中先是定义了一个对象 obj，执行构造函数使得静态数据成员 count 变为了 1，然后又定

义了指针对象，执行构造函数后，count 变为了 2，释放指针后，执行析构函数，使得 count 变为了 1。所以本题答案为 A。

36 数据流图中带有箭头的线段表示的是()。

A. 控制流 B. 事件驱动 C. 模块调用 D. 数据流

参考答案: D

参考解析: 数据流图中带箭头的线段表示的是数据流，即沿箭头方向传送数据的通道，一般在旁边标注数据流名。

37 下列程序的运行结果为()。

```
#include <template>

class FF
{
    TT a1, a2, a3; public:
    FF(TT b1, TT b2, TT b3){a1=b1;a2=b2;a3=b3;} T Sum() {return a1+a2+a3;}
}

void main()
{
    FFx(int(1.1), 2, 3), Y(int(4.2), 5, 6); cout<
}

A. 6.1 15.2 B. 6 15 C. 6 15.2 D. 6 1 15
```

参考答案: B

参考解析: 本题考查模板函数的应用，表达式 int(1.1)和 int(4.2)足将 1.1 和 4.2 强制转换为整型。

38 下列选项中不属于结构化程序设计原则的是()。

A. 可封装 B. 自顶向下 C. 模块化 D. 逐步求精

参考答案: A

参考解析：结构化程序设计的思想包括：自顶向下、逐步求精、模块化、限制使用 goto 语句，所以选择 A。

39 有三个关系 R，S 和 T 如下：

R			S			T		
A	B	C	A	B	C	A	B	C
a	1	2	d	3	2	a	1	2
b	2	1				b	2	1
c	3	1				c	3	1
						d	3	2

其中关系 T 由关系 R 和 s 通过某种操作得到，该操作为()。

A. 选择 B. 投影 C. 交 D. 并

参考答案：D

参考解析：并关系 T 中包含了关系 R 与 s 中的所有元组，所以进行的是并的运算。

40 下列控制格式输入/输出的操作符中，能够设置浮点数精度的是()。

A. setprecision B. setc. setfill D. showpoint

参考答案：A

参考解析：iomanip 中定义的操纵符包括：setw(w) 用 ch 填充空白、setprecision(n) 将浮点精度置为 n、setw(w)、读写 w 个字符的值 showpoint 设置小数点。

二、基本操作题

41 请使用 VC6 或使用【答题】菜单打开考生文件夹 proj1 下的工程 proj1，此工程中包含了类 Pets(“宠物”) 和主函数 main 的定义。程序中位于每个“//ERROR****found ****”之后的一行语句有错误，请加以改正。改正后程序的输出结果应为：

Name: Sony Type: dog

Name: John Type: dog

Name: Danny Type: cat

Name: John Type: dog

注意：只修改每个“//ERROR ****found ****”下的那一行，不要改动程序中的其他内容。

```
#include

using namespace sm; enum Pets_type{d09, cat, bird, fish};

class Pets{ private:

char *name; Pets_type type; public:

Pets(const char *name=" sonny" , Pets_type type=dog);

Pets&operator=(const Pets&s);

~Pets();

void show()const;};

Pets: : Pets(econst char$name, Pets_type type)

//构造函数 {

This ->name=new char[strlen(name)+1];

strcpy(this->name, name);

//ERROR ****found**** type=type;

} Pets: : ~Pets()//析构函数，释放 name 所指向的字符串

{ //ERROR ****found***** name=' /0 ' ;

}

Pets&Pets: : Operator=(const Pets&s){

if(&s==this)//确保不要向自身赋值 return *this;

delete[]name;
```

```
name=new char[strlen(S.name)+1];//ERROR *****found*****
```

```
strcpy(this->nmne, name); type=S.type;
```

```
return *this;}
```

```
void Pets: : show()const
```

```
cout<< "Name: " <
```

参考解析:

(1) this->type=type;

(2) delete[] name;

(3) strepy(this->name, s.name);

三、简单应用题

42 请使用 VC6 或使用【答题】菜单打开考生文件夹 proj 下的工程 proj2, 该工程中包含一个程序文件 main.Cpp, 其中有坐标点类 point, 线段类 Line 和三角形类 Triangle 的定义. 还有 main 函数的定义。程序中两

点间距离的计算是按公式 $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 实现的, 三角形面积的计算是按公式 $f = \sqrt{s(s-a)(s-b)(s-c)}$ 实现的, 其中 $s = \frac{a+b+c}{2}$ 。请在程序中的横线处填写适当的代码, 然后删除横线, 以

Side 1: 9.4339 Side 2: 5

Side 3: 2 area: 20 注意: 只在横线处填写适当的代码, 不要改动程序中的其他内容, 也不要删除或移动 “//****found****”。

```
#include #nClude using namespaCe std;
```

```
Class Point{//坐标点类 public:
```

```
Const double x, y;
```

```

Point(double x=0.0, double Y=0.0): x(x), Y(Y) {} //*****found*****

double distanceTo(____) Const{

//到指定点的距离

return sqrt((x-P.X)*(x-P.x)+(Y-P.Y)*(Y-P.Y)); }

};

Class Line{//线段类 public:

    const Point p1, p2;//线段的两个端点

//*****found*****

    Line(Point p1, Point p2): ____{

double length()const{return p1.distanceTo(p2);} //线段的长度

};

Class Triangle{//三角形类 public:

    const Point p1, p2, p3;//三角形的三个顶点 //*****found*****

    Triangle(____): p1(p1), p2(p2), p3(p3) {} double length1()Const{//边 p1,
p2 的长度

return Line(p1, p2).length(); }

double length2()Const{//边 p2, p3 的长度 return Line(p2, p3).length();

}

double length3()Const{//边 p3, p1 的长度 return Line(p3, p1).length();

}

double area()Const{//三角形面积

//*****found***** double s=____;

```

```

return sqrt(s{(S-length1())*(s-length2())*(s-length3())}); }

};

int main() {

    Triangle r(Point(0.0, 8.0), Point(5.0, 0.0), Point(0.0, 0.0)); Cout<<"
Side 1: " <

    Cout<<" Side 2: " <

    return 0;
}

```

参考解析:

(1) const Point&P

(2) p1(p1), p2(p2)

(3) Point p1, Point p2, Point p3

(4) (length1()+length2()+length3())/2

四、综合应用题

43 请使用 VC6 或使用【答题】菜单打开考生文件夹 proj3 下的工程 proj3，其中声明的 DataList 类，是一个用于表示数据表的类。DataList 的重载运算符函数 operator+，其功能是求当前数据表与另一个相同长度的数据表之和；即它返回一个数据表，其每个元素等于相应两个数据表对应元素之和。请编写这个 operator+ 函数。程序的正确输出应该是：

两个数据表：1, 2, 3, 4, 5, 6 3, 4, 5, 6, 7, 8 两个数据表之和：

4, 6, 8, 10, 12, 14 要求：

补充编制的内容写在“//*****333*****”与“//*****666*****”之间，不得修改程序

的其他部分。

注意：程序最后将结果输出到文件。ut.dat 中。输出函数 writeToFile 已经编译为。bj 文件，并且在本程序中调用。

```
//DataList.h

#include using namespace std;

Class DataList{//数据表类 intfen;

double*d; public:

DataList(int len, double data[]=NULL); DataList(DataList&data);

int length()Const{return len;}

double getElement(int i)Const{return d[i];}

DataList operator+(Const DataList&list)Const;//两个数据表求和 void
show()Const;//显示数据表

};

void writeToFile(Char$, Const DataList*); //main.Cpp

#include "DataList.h"

DataList::DataList(int len, double data[]): len(len){ d=new double[len];
for(int i=0;i
d[i]=(data==NULL?0.0: data[i]);
}

DataList::DataList(DataList&data): len(data.len){ d=new double[len];
for(int i=0;i

DataList DataList::operator+(Const DataList&list, )Const{//两个数据表
求和 double*dd=new double[list.length()];
```

```

//*****333*****

//*****666*****

return DataList(list.length(), dd); }

void DataList::show()Const{//显示数据表 f()r(int。i=0;i

eout<

Cout<

int main(){

double s1[]={1, 2, 3, 4, 5, 6};

double s2[]={3, 4, 5, 6, 7, 8};

DataList list1(6, s1), list2(6, s2);//定义两个数据表对象 eout<<" 两个数
据表: " <

list1.show(); list2.show(); Cout<

(list1+list2).show();

writeToFile(" ", list1+list2);return 0;

参考解析:

for(int i=0;i

dd[i]=d[i]+list.d[i];

```