

一、选择题（每题 1 分，共 10 题，共 10 分）

- 1、在数组 `int b[][4]={ {1},{3,4},{4,7,9,10},{8,5,6}}` 中，`b[2][2]` 的值是 D 。
A. 0 B. 4 C. 7 D. 9
- 2、以下关于 `this` 指针描述正确的是 C 。
A. 使基类公有成员在子类中可以被访问。
B. `this` 指针必须写成显式的。
C. `this` 指针总指向要调用的其成员函数的对象。
D. 静态成员函数也存在 `this` 指针。
- 3、如果通过 `new` 运算符动态分配失败，返回结果是 B 。
A. -1 B. 0 C. 1 D. 不确定
- 4、 D 是一种限制存取位置的线性表，元素的存取必须服从先进先出的规则。
A. 顺序表 B. 链表 C. 栈 D. 队列
- 5、下列关于指针运算的描述，错误的是 D 。
A. 可将一个空指针赋给某个指针。
B. 两个指针在一定条件下，可以进行相等和不等运算。
C. 一个指针可以加上一个整数，指向当前元素后面的若干个位置的元素。
D. 两个指针在一定条件下可以相加。
- 6、实现深复制，下面的类成员函数中， A 不是必须自定义的。
A. 构造函数 B. 复制构造函数
C. 析构函数 D. 复制赋值操作符函数
- 7、设数组 `int fibon[10]`，`int *pfib=fibon`；则访问 `fibon` 数组第二个元素，以下写法错误的是 B 。
A. `fibon[1]` B. `*++fibon` C. `*++pfib`
D. `*(pfib+1)`
- 8、假设 `Person` 类包含公有成员 `name`，私有成员 `id` 和受保护成员 `GetID`，而 `Student` 类私有继承了 `Person` 类，那么 `Student` 类的成员函数可以直接访问 C 。
A. `Person` 类的所有成员

- B. 仅有 Person 类的公有成员 name
- C. 仅有 Person 类的公有成员 name 和受保护成员 GetID
- D. Person 类的任何成员都不能访问

9、实现多态的派生类函数不必 D。

- A. 与基类虚函数同名
- B. 与基类虚函数同参数表
- C. 与基类虚函数同返回类型
- D. 用 virtual 修饰

10、分析下列代码是否存在问题，选择合适的选项： B 。

```
int main(void)
{
    int *p = new int [10];
    p = new int [10];
    delete [] p;
    p = NULL;
    return 0;
}
```

- A. 没有问题
- B. 有内存泄漏
- C. 存在空悬指针
- D. 存在重复释放同一空间

二、填空题（每空 1 分，共 20 空，共 20 分）

1、有序数组 int B[17]中存放 17 个元素，用对半查找法查找 B[11]元素，则进行比较的数组下标值依次是 8,12,10,11。

2、设整数型指针 P1，P2 分别指向整数型数组 A[10]={1,2,0,4,5,9,7,8,6,4}的第 2 和第 5 个元素，则 P2-P1=3，A[5]-A[2]=9。

3、单链表的结点分为 指针 域和 数据 域两部分。

4、标明为无具体实现的虚函数是 纯虚函数。包含该函数的类称为 抽象类，不能用来定义对象。

5、C++文件流采用两种格式访问文件：文本格式和二进制格式。前者按 字符 存取，后者按 字节 存取。

6、重载提取运算符>>和插入运算符<<实现对象的输入和输出，需要将重载的运算符函数声明为该类的友元函数。

7. 指针类型变量用于存贮地址，在内存中它占有 4 个存贮单元。

8. 设有说明：

```
int a, k, m, *p1=&k, *p2=&m;
```

执行 `a=p1==&m;`后 a 的值是 0。

9. 若有：

```
int i,&j=i;
```

```
i=1;
```

```
j=i+2;
```

则 `i=` 3。

10. 构造函数 是一种特殊的成员函数，它主要用来为对象分配内存空间，对类的数据成员进行初始化并执行对象的其他内部管理操作。

11. 一般情况下，使用系统提供的默认析构函数就可以了，但当对象的成员中使用了new运算符动态分配内存空间时，就必须定义析构函数以正确释放对象空间。为了对象间能正确赋值，还必须要重载运算符=。

12. 在类的派生过程中，要实现动态多态性，首先在类中必须要定义虚函数，还要在使用对象的函数中定义基类指针，使该指针指向不同类的对象。

13. 利用成员函数对双目运算符重载，其有1个参数，该参数为运算符的右操作函数。

三、阅读程序题（每空 1-2 分，共 40 分）

1、以下程序的输出结果是：（本题 6 分，每空 2 分）

```
#include <iostream>
using namespace std;
```

```

void main(void)
{
    int a,b,k=4,m=6,*p1=&k,*p2=&m;
    int arr[]={30,25,20,15,10,5},*p=arr;
    p++;
    cout<<*(p+3)<<endl;
    a=(p1==&m);
    b=(*p1)/(*p2)+7;
    cout<<"a="<<a<<endl;
    cout<<"b="<<b<<endl;
}

```

程序运行结果：

```

      10
-----
      a=0
-----
      b=7
-----

```

2、改正以下程序的错误：（本题 8 分，每空 2 分）

```

#include <iostream>
using namespace std;
class Sample
{
    int value;
public:
    void Sample( int a ) { value =a;}
    int Max (int x,int y) {return x>y?x:y;}
    int Max (int x,int y,int z=0)
    {
        if (x>y)
            return x>z?x:z;
        else
            return y>z?y:z;
    }
    ~Sample (int a) {value =0;}
};
void main(void)
{
    Sample s(4);
    cout<<s.value<<endl;
    s.Max(10,20);
}

```

以上程序中的错误有：

```

Void 去掉
-----
重载产生二义性
-----

```

析构函数不能有参数

私有数据成员不可类外访问

3、指出程序的运行结果：（本题 8 分，每空 2 分）

```
#include <iostream>
using namespace std;
class Vector
{
    int x,y;
public:
    Vector() { };
    Vector(int i,int j) {x=i;y=j;}
    void disp() {cout << "("<<x<<","<<y<<")"<<endl;}
    void operator+=(Vector D) {x+=D.x; y+=D.y;}
    void operator-=(Vector D) {x-=D.x; y-=D.y;}
};
void main(void)
{
    Vector A(1,2),B(4,2);
    cout<<"A:";
    A.disp( );
    cout<<"B:";
    B.disp( );
    A+=B;
    cout<<"A+=B:";
    A.disp( );
    A-=B;
    cout<<"A-=B:";
    A.disp( );
}
```

程序运行结果：

A(1,2)

B (4,2)

A+=B(5,4)

A-=B(1,2)

4、下面是一个实现类的继承与派生的程序，请写出程序运行结果，并根据主函数中编程者的原意(调用派生类的成员函数)，修改类的成员定义，然后写出修改后的运行结果（本题 10 分）

```
#include <iostream>
using namespace std;
class A
```

```

{
public:
    virtual void fun1() {cout <<"A fun1"<<endl;}
    virtual void fun2() {cout <<"A fun2"<<endl;}
    void fun3() {cout <<"A fun3"<<endl;}动态绑定啦
};
class B:public A
{
public:
    void fun1() {cout <<"B fun1"<<endl;}
    void fun2(int x) {cout <<"B fun2"<<endl;}
    void fun3() {cout <<"B fun3"<<endl;}
};
void main(void)
{
    A *p;
    B b;
    p=&b;
    p->fun1( );
    p->fun2( );
    p->fun3( );
}

```

修改前输出结果如下（每空 1 分）：

```

B fun1
B fun2
A fun3

```

纠正错误：（每空 2 分）

```

Fun3 改为虚函数
Fun2 参数 X 去掉

```

修改后的输出结果如下（每空 1 分）：

```

B fun1
B fun2
B fun3

```

5、指出程序的运行结果：（本题 8 分，每空 1 分）

```

#include <iostream>
using namespace std;
class B1
{
public:
    B1() {cout<<"B1:Constructor"<<endl;}
    ~B1() {cout<<"B1:Destructor"<<endl;}
};
class B2

```

```

{
public:
    B2() {cout<<"B2:Constructor"<<endl;}
    ~B2() {cout<<"B2:Destructor"<<endl;}
};
class B3
{
public:
    B3() {cout<<"B3:Constructor"<<endl;}
    ~B3() {cout<<"B3:Destructor"<<endl;}
};
class A:public B2,public B3
{
    B1 b1;
public:
    A():B3(),B2(),b1() {cout<<"A:Constructor"<<endl;}
    ~A() {cout<<"A:Destructor"<<endl;}
};
void main(void)
{
    A a;
}

```

先把继承的搞掉，然后是类里面的，然后是自己。先基类再派生类
该程序的执行结果如下：

```

B2      Constructor_____
B3      Constructor_____
B1      Constructor_____
A       Constructor_____
A       Destructor_____
B1      Destructor_____
B2      Destructor_____
B3      Destructor_____

```

四、完善程序填空题（每空 2 分，共 15 空，共 30 分）

1. 完成如下程序，要求实现方阵（矩阵行列数目相同）的上三角元素（含对角线元素）的和。

```

#include<iostream>
using namespace std;
const int n=10;
int main(void)
{
    int elements[n][n]= {1,1,1, } _____; /*初始化二维数组，首行前三个
元素为 1，其他元素为*/

```

```

    int sum=0;
    for(int i=0;i<n;i++)
        for( int j=i; j<n; j++)
            sum+=elements[i][j];
    cout<<sum<<endl; /*输出上三角的和*/
    return 0;
}

```

2. 如下已定义点类 Point，包含 x，y 坐标数据成员；再采用聚合和派生两种复合方式定义线段类 Line。完成 Line 类的定义与实现。

```

class Point
{
    friend class Line;
protected:
    double x, y;
public:
    Point(double xv = 0, double yv = 0) {x = xv; y = yv;}
    double area() {return 0;}
    void show() {cout<<"点坐标: x="<<x<<' ' <<"y="<<y<<endl;}
};

class Line : public Point
{
    Point end; //终点
public:
    Line(double xv1 = 0, double yv1 = 0, double xv2 = 0, double yv2 = 0) :
        Point(xv1, yv1), end(xv2, yv2) { }
    double getLength()
    {
        return sqrt((x - end.x) * (x - end.x) + (y - end.y) * (y - end.y));
    }
    double area() {return 0;}
    void show()
    {
        cout<<"起点:  \n";
        Point::show();
        cout<<"终点:  \n";
        end.show();
    }
};

```

3. 下列程序将结构体变量 tt 中的内容写入 D 盘上的 date.txt 文件。


```

#include <fstream.h>
#include < stdlib.h >
struct date{
    int year, month, day;
};
void main(void) {
    date tt={2009, 6, 10};

    ofstream  _____;

    outdate.open("d:\\date.txt", ios::binary);

    if ( _____ )

        { cerr << "\n D:\\writel.dat not open " << endl ;
          exit (-1) ; }

    outdate.write( _____, sizeof(tt));

    _____ outdata.close();

}

```

4. 完善 fruit 类的构造函数、析构函数、深拷贝构造函数

```

#include<iostream>
#include<string>
using namespace std;
class fruit
{
    string name; //水果名称
    string shape; //水果形状
    string color; //水果颜色
    double weight;//水果均重
    double *price;//指向一年中每个季度水果的均价
public:
    fruit( string N, string S, string C, double W, double *p=0):
        name(N), shape(S), color(C), weight(W), price(P) {}
    ~fruit()
    {
        if(price) {delete _____;}
    }
    fruit(fruit& F)
    {
        name=F.name;

```

```

        shape=F.shape;
        color=F.color;
        weight=F.weight;
        if(F.price)
        {
            price= new double[4];
            for(int i=0;i<4;i++)
                price[i]=F.price[i];
        }
        else price=F.price;

    }
    void setprice()
    {
        if(price)
        {
            cout<<"请输入每个季度"<<name<<"的均价: \n";
            for(int i=0;i<4;i++)
                cin>>price[i];
        }
    }
};

int main(void)
{
    double *price_melon=new double[4];
    fruit strawberry ;
    //运行后 strawberry 中的数据成员依次为: #, #, #, 0, NULL
    Fruit melon("watermelon","sphere","black and green",8.5,price_mel
on);
    melon.setprice();
    fruit Dongtai_melon(melon);
    return 0;
}

```

5. 下面是一个选择排序函数，完成将数组元素值按上升的次序来排序，请完善程序。

```

void sort(int a[],int n){
    for (int i = 0; i < n; i++){
        int pos=i;
        for (int j= i+1; j < n; j++){
            if (a[j]<a[pos]) pos=j;
        }
    }
}

```

```
        int temp = a[pos];  
        a[pos] = a[i];  
        a[i] = temp;  
    }  
}
```