



# 南 卷 汇

2016 大一下面向对象程序设计期中试题汇

南洋书院学生会

制作

## 目录

面向对象程序设计期末 1 .....	2
面向对象程序设计期末 2 .....	10
编程题.....	20
面向对象程序设计 1 参考答案 .....	23
面向对象程序设计 2 参考答案 .....	24
编程题答案 .....	25
面向对象程序设计期末 3 .....	34
面向对象程序设计期末 3 参考答案 .....	43
面向对象程序设计期末 4 .....	45
面向对象程序设计期末 4 参考答案 .....	54
“计算机程序设计” 机考试题 .....	60
“计算机程序设计” 机考试题答案 .....	62

# 面向对象程序设计期末 1

制作人：唐龙

## 一、单项选择题

1. C++对 C 语言有很多改进，下列描述中\_\_\_\_\_使得 C 语言发生了质变，即从面向过程变为面向对象。

- A. 增加了一些新的运算符
- B. 允许重载，并允许设置缺省值
- C. 函数说明必须用原型
- D. 引进了类和对象的概念

2. 设置虚基类的目的为\_\_\_\_\_

- A. 简化程序
- B. 消除二义性
- C. 提高运行效率
- D. 减少目标代码

3. 下面概念不正确的是

- A. 构造函数的名字必须与类名相同
- B. 构造函数、析构函数均无返回值
- C. 析构函数可以被重载
- D. 析构函数可以成为虚拟函数

4. 关于 this 指针不正确的是\_\_\_\_\_

- A. this 指向类对象的首地址
- B. this 不能被对象外部所引用
- C. this 可被该类的所有成员函数所引用
- D. this 不可被友元函数所引用

5. 下列关于运算符重载的正确概念是\_\_\_\_\_

- A. 所有运算符都可以被重载
- B. C++中可定义新的运算符
- C. 重载运算符不能改变原有运算符的参数个数、优先级、结合性
- D. 凡是能被成员函数重载的运算符也一定可被友元函数重载

6. 某类 A 中静态成员函数 display 的原型如下：`static void display();` 则下列说法正确的是\_\_\_\_\_

- A. 该函数可直接使用类 A 的普通数据成员

B.该函数可使用 this 指针

C.该函数只能使用类 A 的静态成员

D.该函数既可使用类 A 的普通成员函数也可使用类 A 的静态成员函数

7.下列不属于同一函数原型的是\_\_\_\_

A.add(int, double)

B.int add(int, double)

C.int add(int x, double y)

D.int add(double y, int x)

8.在下列重载函数的调用时可能发生错误的是\_\_\_\_

A.int print(int x); void print(float);

B.int show(int x); int show(char\* s);

C.int display(Myclass A); char\* display(Myclass A);

D.int view(int x, int y); int view(int x);

9.下列哪个初始化是非法的\_\_\_\_，其中后面选项中的变量类型默认为前面选项出现过的类型。

A.int i = -1;

B.const int ic = i;

C.const int\* pc = &ic;

D.int\* const cpic = &ic;

10.有如下定义语句:

```
int* array_ptr = new int[10];
```

```
int* pi;
```

执行下面哪个语句将使程序执行 delete[] array\_ptr 时出错\_\_\_\_

A.(\*array\_ptr)++;

B.array\_ptr--;

C.pi = array\_ptr;

D.pi = array\_ptr + 1;

## 二、填空题

1.已知下列类层次结构，他们都定义了缺省构造函数

```
class A {...};
```

```
class B: public A {...};
```

```
class C: public B {...};
```

```
class X {...};
```

```
class Y {...};
```

```
class Z: public X, public Y {...};
```

```
class MI: public C, public Z {...};
```

定义 MI mi 的构造函数的执行顺序是\_\_\_\_\_

```
2. #include <iostream>
using namespace std;
int c;
int& funny(int& a, int b)
{
    if (a != b)
        c = a + b;
    a = b;
    b = c;
    return c;
}
int main()
{
    int x = 3, y = 4, z = 5;
    cout << (funny(x, y) = z) << endl;
    cout << "x is " << x << ", and y is " << y << endl;
    z = funny(x, y);
    cout << "x is " << x << ", and y is " << y << ", and z
is " << z << endl;
    return 0;
}
```

程序的运行结果为:\_\_\_\_\_

```
3. #include <iostream>
using namespace std;
class base {
public:
    virtual void func1() {
        cout << "base func1" << endl;
    }
    virtual void func2() {
        cout << "base func2" << endl;
    }
    void func3() {
        cout << "base func3" << endl;
    }
};
```

```
class derived :public base {
public:
    void func1() {
        cout << "derived func1" << endl;
    }
    void func2(int x) {
        cout << "derived func2" << endl;
    }
    void func3() {
        cout << "derived func3" << endl;
    }
};

int main() {
    derived d;
    base& bp = d;
    bp.func1();
    bp.func2();
    bp.func3();
    return 0;
}
```

程序的运行结果为: \_\_\_\_\_

```
4. #include <iostream>
using namespace std;
int num = 10;
class one_class {
    int num;
public:
    void set1(int i) {
        num = i;
    }
    void set2(int& i) {
        int num = 0;
        i = num;
    }
    int get() {
        return num;
    }
    ~one_class() {
        ::num = num;
    }
};

int main() {
```

```

    one_class obj;
    obj.set1(100);
    obj.set2(num);
    cout << num << endl;
    cout << obj.get() << endl;
    cout << num << endl;
    return 0;
}

```

程序的运行结果为: \_\_\_\_\_

```

5. #include <iostream>
using namespace std;
class A {
    int i;
public:
    A(int temple) :i(temple) {
        cout << "A constructing\t" << i << endl;
    }
};
class B {
public:
    B() {
        cout << "B constructing." << endl;
    }
};
class C :public A, public B {
public:
    C(int temple) :A(temple), B() {
        cout << "C constructing." << endl;
    }
};
class D :virtual public C {
    int i;
public:
    D(int templ, int tempt) :C(templ), i(tempt) {
        cout << "D constructing." << endl;
    }
};
class E :virtual public C {
public:
    E(int templ) :C(templ) {
        cout << "E constructing." << endl;
    }
}

```

```
};
class F :public D, public E {
    A a;
public:
    F(int i, int j, int k, int l, int m) :E(i), D(j, k),
    C(l), a(m){}
};
int main() {
    F f(1, 2, 3, 4, 5);
    return 0;
}
```

程序的运行结果为: \_\_\_\_\_

### 三、阅读程序 填写代码

```
1. #include <iostream>
#include <string.h>
class string {
    char* str;
public:
    string() {
        str = new char[1];
        *str = '\0';
    }
    string(char* s) {
        if (s) {
            _____
        }
        else {
            str = new char[1];
            *str = '\0';
        }
    }
    string(const string& obj) {
        char* s = "copy from";
        str = new char[strlen(s) + 1];
        strcpy(str, s);
        strcpy(str + strlen(s), obj.str);
    }
    string& operator=(const string& obj) {
        _____
        delete[] str;
```



```

        str = new char[strlen(obj.str) + 1];

        return *this;
    }
    void print() {
        std::cout << str << std::endl;
    }
    ~string() {
        delete[] str;
    }
};
int main() {

    string obj3;

    obj1.print();
    obj2.print();
    obj3.print();
}

```

程序的运行结果为：

world

copy from world

world

#### 四、找出不能通过编译的语句，并解释原因

```

1  #include <iostream>
2  using namespace std;
3  class base{
4      int base_i;
5  protected:
6      char base_c;
7  public:
8      float base_f;
9      base(int i, char c, float f) :base_i(i),
10     base_c(c), base_f(f) {}
11     void base_set(int i, char c, float f) {
12         base_i = i;
13         base_c = c;
14         base_f = f;
15     }

```

```

16 };
17 class derived :protected base {
18     int derived_i;
19 protected:
20     char derived_c;
21 public:
22     float derived_f;
23     derived(int i, char c, float f) :base(i, c, f),
24     derived_i(i), derived_c(c), derived_f(f) {}
25     void derived_set(int i, char c, float f) {
26         base_i = i;
27         derived_i = i;
28         base_c = c;
29         derived_c = c;
30         base_f = f;
31         derived_f = f;
32     }
33 };
34 struct final:derived {
35     static int count = 0;
36     final(int i, char c, float f): derived(i, c, f){}
37     void final_set(int i, char c, float f) {
38         base_c = c;
39         derived_c = c;
40         base_f = f;
41         derived_f = f;
42     }
43 };
44 int main() {
45     final fobj;
46     fobj.derived_set(1, 1, 1);
47     fobj.base_set(1, 1, 1);
48     return 0;
49 }

```

## 五、简答题

1. 友元运算符函数和成员运算符函数有什么不同？
2. 什么是纯虚函数？什么是抽象类？定义抽象类的作用是什么？

## 六、编程题

1. 有下列主函数：

```
int main() {  
    array<int> array_int(s);  
    array<char> array_char;  
    array<float> array_float(10);  
}
```

请编写一个能够处理上述简单数据类型的 array 类模板，在该类模板中只需要写出必要的数据成员和构造函数即可；附：写出析构函数。

2. 给出类

```
class three_d {  
    int x, y, z;  
public:  
    three_d(int i, int j, int k) :x(i), y(j), z(k) {}  
    three_d() :x(0), y(0), z(0) {}  
};
```

针对给出的类，重载 "\*" 运算符，使 "\*" 运算符可以接受下面的运算形式：

ob\*int, int\*ob; 附：本科生只需完成上面运算的任意一种即可。

## 面向对象程序设计期末 2

制作人：李源勋

### 第一部分 单项选择题

一、单项选择题 (本大题共 10 小题，每小题 2 分，共 20 分) 在每小题列出的四

个选项中只有一个选项是符合题目要求的，请将其代码填在题后的括号内。错选或未选均无分。

1. 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

2. 在 C++ 程序中，对象之间的相互通信通过 ( )

- A. 继承实现
- B. 调用成员函数实现
- C. 封装实现
- D. 函数重载实现

3. 对于任意一个类，析构函数的个数最多为 ( )

- A. 0
- B. 1
- C. 2
- D. 3

4. 下面函数模板定义中不正确的是 ( )

- A. 

```
template<class Q>
QF(Q x){
    return Q+x;
}
```
- B. 

```
template<class Q>
QF(Q x){
    return x+x;
}
```
- C. 

```
template<class T>
TF(T x){
    return x*x;
}
```
- D. 

```
template<class T>
TF{T x){
    return x>1;
}
```

5. 友元关系不能 ( )

- A. 继承
- 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. 提高程序的运行效率

6. 语句 `ofstream f("SALARY.DAT", ios::app | ios::binary);` 的功能是建立流对象 `f`，试图打开文件 `SALARY.DAT` 并与之连接，并且 ( )

- A. 若文件存在，将文件写指针定位于文件尾；若文件不存在，建立一个新文件
- B. 若文件存在，将其置为空文件；若文件不存在，打开失败

C. 若文件存在，将文件写指针定位于文件首；若文件不存在，建立一个新文件

D. 若文件存在，打开失败；若文件不存在，建立一个新文件

7. 下面说法正确的是 ( )

A. 内联函数在运行时是将该函数的目标代码插入每个调用该函数的地方

B. 内联函数在编译时是将该函数的目标代码插入每个调用该函数的地方

C. 类的内联函数必须在类体内定义

D. 类的内联函数必须在类体外通过加关键字 `inline` 定义

8. 可以用 `p.a` 的形式访问派生类对象 `p` 的基类成员 `a`，其中 `a` 是 ( )

A. 私有继承的公有成员

B. 公有继承的私有成员

C. 公有继承的保护成员

D. 公有继承的公有成员

9. 在公有派生情况下，有关派生类对象和基类对象的关系，不正确的叙述是 ( )

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

10. 对于类定义

```
class A{
public:
    virtual void func1(){}
    void func2(){}
};
class B:public A{
public:
    void func1(){
        cout << "class B func1" << endl;
    }
}
```

```
virtual void func2(){
    cout << "class B func2" << endl;
}
};
```

下面正确的叙述是 ( )

- A. A::func2() 和 B::func1() 都是虚函数
- B. A::func2() 和 B::func1() 都不是虚函数
- C. B::func1() 是虚函数，而 A::func2() 不是虚函数
- D. B::func1() 不是虚函数，而 A::func2() 是虚函数

## 第二部分 非选择题 (共 80 分)

二、填空题 (本大题共 10 小题，每小题 2 分，共 20 分) 不写解答过程，将正确的答案写在每小题的横线处，错填或不填均无分。

11. 定义类的动态对象数组时，系统只能够自动调用该类的\_\_\_\_\_构造函数对其进行初始化。
12. 无论是什么继承方式，派生类的成员不能访问基类\_\_\_\_\_属性的成员。
13. 表达式 `cout<<endl` 还可表示为\_\_\_\_\_
14. 基类的公有成员在派生类中的访问权限由\_\_\_\_\_决定。
15. C++ 支持的两种多态性分别是\_\_\_\_\_多态性和\_\_\_\_\_多态性。
16. C++ 中语句 `const char* const p="hello";` 所定义的指针 `p` 和它所指的内容都不能被\_\_\_\_\_
17. 假定 `AB` 为一个类，则语句 `AB (AB& x);` 为该类的\_\_\_\_\_函数的原型说明。
18. 说明常成员函数时，关键字 `const` 写在成员函数的\_\_\_\_\_和\_\_\_\_\_之间。
19. 在 C++ 中，访问一个对象的成员所用的运算符是\_\_\_\_\_，访问一个指针所指向的对象的成员所用的运算符是\_\_\_\_\_
20. 派生类从一个或多个以前定义的该类的\_\_\_\_\_继承数据和函数。

三、改错题(本大题共 5 小题，每小题 2 分，共 10 分)

21.指出下面程序段中的错误，并说明出错原因。

```
class A{
    int a,b;

public:
    A(int aa,int bb){
        a=aa;
        b=bb;
    }
};
    A x(2,3),y(4);
```

22.指出并改正下面利用类模板的对象定义中的错误。

```
template<class T>
class Tany{
    T x,y;
public:
    Tany(T a,T b){
        x=a;
        y=b;
    }
    T sum(){
        return x+y;
    }
};
Tany(int) obj(10,100);
```

23.指出下面程序段中的错误，并说明出错原因。

```
class one{
private:
    int a;
public:
    void func(two&);
};
```

```
class two{
private:
    int b;
    friend void one::func(two&);
};
void one::func(two& r){
    a=r.b;
}
```

24.指出下面程序段中的错误，并说明出错原因。

```
#include <iostream>
class A{
public:
    void fun(){
        cout << "a.fun" << endl;
    }
};
class B{
public:
    void fun(){
        cout << "b.fun" << endl;
    }
    void gun(){
        cout << "b.gun" << endl;
    }
};
class C:public A,public B{
private:
    int b;
public:
    void gun(){
        cout << "c.gun" << endl;
    }
    void hun(){
        fun();
    }
}
```



```
};
```

25. 指出下面程序段中的错误，并说明出错原因。

```
class Location{
    int X,Y=20;
protected:
    int zeroX,zeroY;
    int SetZero(int ZeroX,int ZeroY);
private:
    int length,height;
public:
    float radius;
    void init(int initX,int initY);
    int GetX();
    int GetY();
};
```

四、完成程序题(本大题共 5 小题，每小题 4 分，共 20 分)

根据题目要求，完成程序填空。

26. 在下面横线处填上适当字句，完成类中成员函数的定义。

```
class A{
    int* a;
public:
    A(int aa=0){
        a=_____ ;
    }
    ~A() { _____ }
};
```

27. 下面是一个函数模板，用于计算两个向量的和。在下面横线处填上适当字句，完成函数模板定义。

```
#include <iostream>
template<class T>
T* f(T* a,T* b,int n)
{
    T* c=_____ ;
```

```

    for(int i=0;i<n;i++)
        c[i]=_____;;
    return c;
}
void main()
{
    int a[5]={1,2,3,4,5},b[5]={10,20,30,40},*p;
    p=f(a,b,5);
    for(int i=0;i<5;i++)
        cout<<p[i]<<endl;
}

```

28.下面是一个用户口令检查程序，在横线处填上适当语句完成程序功能。

```

#include <iostream>
#include <string.h>

_____;//定义由 PASS 指针所指向的口令 wolleh。

void main()
{
    char user[10];//用于存放输入的口令
    cout<<"please input your password:"<<endl;
    _____;//输入 口令
    if((strcmp(user,PASS))==0)
        cout<<"your password is correct"<<endl;
    else
        cout<<"your password is error"<<endl;
}

```

29.下面是类 fraction(分数)的定义，其中重载的运算符<<以分数形式输出结果，例如将三分之二输出为 2/3。

在横线处填上适当字句，

```

class fraction{
    int den; //分子

    int num; //分母

```

```

    friend ostream& operator<<(ostream&,fraction);
    ...
};
ostream& operator<<(ostream& os,fraction fr){
    _____;
    return _____;
}

```

30.在下面程序横线处填上适当的字句，使其输出结果为 0,56,56。

```

#include <iostream>
class base{
public:
    _____ func(){return 0;}
};
class derived:public base{
public:
    int a,b,c;
    _____ setValue(int x,int y,int z){a=x;b=y;c=z;}
    int func(){return (a+b)*c;}
};
void main()
{
    base b;
    derived d;
    cout<<b.func()<<','<<'\n';
    d.setValue(3,5,7);
    cout<<pb.func()<<','<<'\n';
    base& pb=d;
    cout<<pb.func()<<endl;
}

```

五、程序分析题(本大题共 6 小题，每小题 5 分，共 30 分)

给出下面各程序的输出结果。

```

31.#include <iostream>
void main()
{

```

```
int *a;
int *&p=a;
int b=10;
p=&b;
cout<<*a;
}
```

输出为: \_\_\_\_\_

```
32.#include <iostream>
template<class T>
Tf(T* a,T* b,int n){
    Ts=(T)0;
    for(int i=0;i<n;i++)
        s+=a[i]*b[i];
    return s;
}
void main()
{
    double c[5]={1.1,2.2,3.3,4.4,5.5};
    double d[5]={10.0,100.0,1000.0};
    cout<<f(c,d,5)<<endl;
}
```

输出为: \_\_\_\_\_

## 编程题

制作人：潘文雯

### 第十二讲补充习题

定义抽象基类 Shape，由它派生出 5 个派生类：Circle（圆形）、Square（正方形）、Rectangle（矩形）、Trapezoid（梯形）、Triangle（三角形）。用虚函数分别计算几种图形面积，并求它们的和。要求用基类指针数组，使它每一个元素指向一个派生类对象。

### 第十三讲补充习题

文件处理，要求：

- （1）当用户从键盘输入某个文件名后，程序可以将该文件内容全文显示在屏幕上；
- （2）程序可以统计该文件中的总字符数；
- （3）程序可以统计该文件中的单词（以空格为单词分隔符）个数；
- （4）显示上面统计出的两个数字。

#### 1.（20 分）

求 100~999 之间所有各位数字的立方和等于 1099 的整数。

编程要求和评分标准：

- （1）能正确取出百位、个位数字；（5 分）
- （2）能正确取出十位数字；（5 分）
- （3）循环结构，分支结构正确；（5 分）
- （4）输出格式正确。（5 分）

#### 2.（20 分）

定义  $5 \times 5$  二维数组，找出第  $m$  行中元素的最大值，并输出最大值元素以及所在的列号。

编程要求和评分标准：

- (1) 数组及变量定义正确 (5 分)
- (2) 键盘输入，并按 5 行 5 列格式输出原数组 (5 分)
- (3) 求最大值及列号功能正确 (5 分)
- (4) 输出格式正确，格式如下： (5 分)

当  $5 \times 5$  数组为  $\begin{pmatrix} 45 & 87 & 32 & 34 & 54 \\ 54 & 56 & 76 & 78 & 66 \\ 46 & 78 & 87 & 23 & 43 \\ 12 & 32 & 44 & 56 & 78 \\ 55 & 66 & 77 & 88 & 99 \end{pmatrix}$  时，如果用户输入  $m=2$  (即第 3

行)，则程序输出：第 3 行的最大值：87，所在列号：3

### 3. (20 分)

编写一个函数，用于去掉字符串尾部的空格符，其原型为：

```
char *mytrim(char *string);
```

其中参数 `string` 为字符串，返回值为指向 `string` 的指针。

并编出主函数进行验证。

编程要求和评分标准：

- (1) 子函数头设计正确；(5 分)
- (2) 子函数功能正确；(5 分)
- (3) 主函数调用的方法正确；(5 分)
- (4) 输出结果正确；(5 分)

### 4: (20 分)

定义类 `Point` 及其子类 `Circle`，`Point` 为点类，`Circle` 类为圆类。

Point 类的数据成员为：

```
double x, y;           //点的 x 和 y 坐标
```

Circle 类从 Point 类继承得到，其增加数据成员为：

```
double radius;        //表示圆的半径
```

Circle 类增加的成员函数有计算圆面积的函数、计算圆心到坐标原点距离的函数，其原型分别为：

```
double Area( );
double Distance( );
```

编程要求和评分标准：

- (1) 给出满足要求的完整的 Point 类的定义及其成员函数；（5 分）
- (2) 给出满足要求的完整的 Circle 类的定义和它的构造函数；（5 分）
- (3) 正确实现 Circle 类的其他成员函数；（5 分）
- (4) 编写主函数，从键盘输入一个圆心坐标及半径，计算并屏幕输出圆的面积和圆心到原点距离（5 分）

## 5. (20 分)

编写一个程序，可以读入一个 C++ 语言的源文件，每一行加上行号后保存到另一个后缀为 print.txt 的文件中。

程序输入输出样例：

```
Please input file name: e1305.cpp
1  // Exercise 13_5
2  #include <iostream.h>
.....
27      return 0;
28  }
```

编程要求和评分标准：

- (1) 程序开始执行时提示：“Please input file name:”，能正确输入文件名；（5 分）
- (2) 文件打开和关闭正确；（5 分）
- (3) 能正确读取文件数据；（5 分）
- (4) C++语言的源文件每一行加上行号后保存到另一个后缀为 print.txt 的文件中。（5 分）

## 面向对象程序设计 1 参考答案

### 一、单项选择题

1.D    2.B    3.C    4.C    5.C    6.C    7.D    8.C    9.D    10.B

### 二、填空题

1. A B C X Y Z MI
2. 5  
x is 4, and y is 4  
x is 4, and y is 4, and z is 5
3. derived func1  
base func2  
base func3
4. 0  
100  
0
5. A coustructing 4  
B coustructing  
C coustructing  
D coustructing  
E coustructing  
A coustructing 5

### 三、阅读程序 填写代码

1. str = new char[strlen(s)+1];



---

```

2. if(this == &obj) return *this;
3. return *this;
4. string obj1("world");
5. string obj2(obj1);
6. obj3 = obj1;

```

#### 四、找出不能通过编译的语句，并解释原因

26 行 `base_i` 为私有成员

35 行 静态数据成员不能在类内初始化

45 行 没有默认的构造函数

47 行 `base_set` 为私有成员

## 面向对象程序设计 2 参考答案

### 一、单项选择题

1.D    2.B    3.B    4.A    5.C | A    6.A    7.B    8.D    9.A    10.C

### 二、填空题

11. 拷贝

12. private

13. cout << "\n";

14. 继承方式

15. 编译   运行

16. 修改

17. 构造

18. 函数头   函数体

19. .   -->

20. 基类

### 三、改错题

21. y(4) 改成 y(4,0)
22. Tany(int) 改成 tany<int>
23. void func(two&) 错误，不能前向申明
24. void hun(){  
    fun();  
}  
中的 fun(); 改成 A::fun(); 或 B::fun();
25. y=20 出错，类定义中不能显式地将类的数据成员初始化

#### 四、完成程序题

26. new int(aa)  
delete a
27. new T[n]  
a[i]+b[i]
28. char\* PASS = "wolleh"  
cin >> user;
29. cout << den << "/" << num << endl;  
os
30. int  
void

#### 五、程序分析题

31. 10
32. 2531

### 编程题答案

#### 第十二讲补充习题答案

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

//定义抽象基类 Shape

class Shape
{
public:

    virtual double area() const =0;    //纯虚函数
```

```
};

//定义 Circle(圆形) 类
class Circle:public Shape
{
protected:
    double radius;    //半径
public:
    Circle(double r)
    {
        radius=r;
    }    //构造函数

    virtual double area() const
    {
        return 3.14159*radius*radius;
    };    //定义虚函数
};

//定义 Square(正方形) 类
class Square:public Shape
{
protected:
    double side;
public:
    Square(double s)
    {
        side=s;
    }    //构造函数

    virtual double area() const
    {
        return side*side;
    }    //定义虚函数
};

//定义 Rectangle(矩形) 类
class Rectangle:public Shape
```

```
{
protected:

    double width,height;    //宽与高
public:
    Rectangle(double w,double h)
    {
        width=w;
        height=h;
    }    //构造函数

    virtual double area() const
    {
        return width*height;
    }    //定义虚函数
};

//定义Trapezoid(梯形)类
class Trapezoid:public Shape
{
protected:

    double top,bottom,height; //上底、下底与高
public:
    Trapezoid(double t,double b,double h)
    {
        top=t;
        bottom=t;
        height=h;
    } //构造函数

    virtual double area() const
    {
        return 0.5*(top+bottom)*height;
    } //定义虚函数
};

//定义Triangle(三角形)类
class Triangle:public Shape
{
```

```

protected:

    double width,height; //宽与高
public:
    Triangle(double w,double h)
    {
        width=w;
        height=h;
    }//构造函数

    virtual double area() const
    {
        return 0.5*width*height;
    }//定义虚函数
};

int main()
{
    Circle circle(12.6); //建立 Circle 类对象 circle

    Square square(3.5); //建立 Square 类对象 square

    Rectangle rectangle(4.5,8.4); //建立 Rectangle 类对象
rectangle
    Trapezoid trapezoid(2.0,4.5,3.2); //建立 Trapezoid 类对
象 trapezoid

    Triangle triangle(4.5,8.4); //建立 Triangle 类对象
    Shape
    *pt[5]={&circle,&square,&rectangle,&trapezoid,&triangle};
    //定义基类指针数组 pt，使它每一个元素指向一个派生类对象

    double areas=0.0; //areas 为总面积

    for(int i=0;i<5;i++)
    {
        cout<<pt[i]->area()<<endl;
        areas=areas+pt[i]->area();
    }

    cout<<"total of all areas="<<areas<<endl; //输出总面积

```

```
    return 0;
}
```

### 第十三讲补充习题答案

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    char ch;
    int letter=0, word=1;
    char filename[30];
    cout<<"Enter a file name:";
    cin>>filename;
    ifstream f1(filename);
    if(!f1)
    {
        cout<<"Cannot open the file."<<endl;
        return 1;
    }
    while(f1)
    {
        f1.get(ch);
        cout<<ch;
        letter++;
        if(ch==' '||ch=='\n')
            word++;
    }
    cout<<endl<<"Total letters: "<<letter;
    cout<<endl<<"Total words: "<<word<<endl;
    return 0;
}
```

1.

```
#include <iostream>
using namespace std;
int main( )
{
    int n, i, j, k;
    for(n=100; n<=999; n=n+1)
    {
```

```

        i = n/100;           // 取出 n 的百位数

        j = (n/10)%10;       // 取数 n 的十位数

        k = n%10;           // 取出 n 的个位数

        if(1099==i*i*i+j*j*j+k*k*k)

            cout << n <<"的各位数字立方和等于 1099, 即 1099="<< i
<<"^3 + "<< j <<"^3 + "<< k <<"^3"<<endl;
        }

        return 0;
    }

```

## 2.

```

#include <iostream>
using namespace std;
int main()
{
    int a[5][5];
    for(int i=0;i<5;i++)
    {
        for(int j=0;j<5;j++)
        {
            cin>>a[i][j];
        }
    }//输入矩阵

    cout<<"当 5*5 的矩阵数组为"<<endl;
    for(int i=0;i<5;i++)
    {
        for(int j=0;j<5;j++)
        {
            cout<<a[i][j]<<" ";
        }
        cout<<endl;
    }//输出矩阵

    int m,x,k;
    cin>>m;

```

```
x=a[m][0];
k=0;
for(int i=0;i<5;i++)
{
    if(x<a[m][i])
    {
        x=a[m][i];
        k=i;
    }
}

cout<<"第"<<m+1<<"行的最大值: "<<x<<"，所在列号: "<<k<<endl;
return 0;
}
```

### 3.

```
#include<iostream>
using namespace std;
char *mytrim(char *string)
{
    char *ptr=string;
    while(*ptr!=0)
        ptr++;
    do
    {
        ptr--;
    }
    while(*ptr==' ');
    * (++ptr)=0;
    return string;
}
int main()
{
    char str[] = "The art of computer programming ";
    cout << "截取前的原始字符串是: [" <<str<<']'<<endl;

    cout << "截取空格后的字符串是: [" <<mytrim(str)<<']'<<endl;
    return 0;
}
```

### 4.

```
#include <iostream>
#include <math.h>
```



```
using namespace std;
class Point
{
protected:
    double x,y;
};
class Circle:public Point
{
    double radius;
public:
    Circle(double a=0,double b=0,double c=0)
    {
        x=a;
        y=b;
        radius=c;
    }
    double distance( )
    {
        return sqrt(x*x+y*y);
    }
    double area( )
    {
        return 3.1415926*radius* radius;
    }
};
int main ( )
{
    double a,b,c;
    cin>>a>>b>>c;
    Circle x(a,b,c);
    cout<<x.area( )<<endl;
    cout<<x.distance( )<<endl;
    return 0;
}
```

## 5.

```
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
int main ( )
{
    char ch, name[20];
```

```

int line = 1;
cout<<"Please input file name: ";
cin>>name;
ifstream filein(name);
ofstream fileout("print.txt");
if(!filein||!fileout)
{
    cout << "Cannot open file!";
    return 1;
}
cout << setw(4) << line << " ";
fileout << setw(4) << line << " ";
while(filein)
{
    filein.get (ch);
    cout << ch;
    fileout << ch;
    if(ch=='\n')
    {
        line++;
        cout << setw(4) << line << " ";
        fileout << setw(4) << line << " ";
    }
}
filein.close( );
fileout.close( );
return 0;
}

```

## 面向对象程序设计期末 3

## 第一部分 单项选择题

一、单项选择题 (本大题共 10 小题，每小题 2 分，共 20 分) 在每小题列出的四个选项中只有一个选项是符合题目要求的，请将其代码填在题后的括号内。错选或未选均无分。

1. 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

2. 在 C++ 程序中，对象之间的相互通信通过 ( )

- A. 继承实现                      B. 调用成员函数实现  
C. 封装实现                      D. 函数重载实现

3. 对于任意一个类，析构函数的个数最多为 ( )

- A. 0                      B. 1                      C. 2                      D. 3

4. 下面函数模板定义中不正确的是 ( )

- A. `template<class Q>`  
    `QF(Q x){`  
        `return Q+x;`  
    `}`  
B. `template<class Q>`  
    `QF(Q x){`  
        `return x+x;`  
    `}`  
C. `template<class T>`  
    `TF(T x){`  
        `return x*x;`  
    `}`  
D. `template<class T>`  
    `TF{T x){`  
        `return x>1;`  
    `}`

5. 友元关系不能 ( )

- A. 继承                      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. 提尚程序的运行效率

6.     语                      句                      ofstream

`f("SALARY.DAT", ios::app | ios::binary);`的功能是建立流对象 `f`，试图打开文件 `SALARY.DAT` 并与之连接，并且（ ）

- A. 若文件存在，将文件写指针定位于文件尾；若文件不存在，建立一个新文件
- B. 若文件存在，将其置为空文件；若文件不存在，打开失败
- C. 若文件存在，将文件写指针定位于文件首；若文件不存在，建立一个新文件
- D. 若文件存在，打开失败；若文件不存在，建立一个新文件

7. 下面说法正确的是（ ）

- A. 内联函数在运行时是将该函数的目标代码插入每个调用该函数的地方
- B. 内联函数在编译时是将该函数的目标代码插入每个调用该函数的地方
- C. 类的内联函数必须在类体内定义
- D. 类的内联函数必须在类体外通过加关键字 `inline` 定义

8. 可以用 `p.a` 的形式访问派生类对象 `p` 的基类成员 `a`，其中 `a` 是（ ）

- A. 私有继承的公有成员
- B. 公有继承的私有成员
- C. 公有继承的保护成员
- D. 公有继承的公有成员

9. 在公有派生情况下，有关派生类对象和基类对象的关系，不正确的叙述是（ ）

- 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

10. 对于类定义

```
class A{
public:
    virtual void func1() {}
```

```
void func2(){}
};
class B:public A{
public:
    void func1(){
        cout << "class B func1" << endl;
    }
    virtual void func2(){
        cout << "class B func2" << endl;
    }
};
```

下面正确的叙述是 ( )

- A. A::func2() 和 B::func1() 都是虚函数
- B. A::func2() 和 B::func1() 都不是虚函数
- C. B::func1() 是虚函数，而 A::func2() 不是虚函数
- D. B::func1() 不是虚函数，而 A::func2() 是虚函数

## 第二部分 非选择题 (共 80 分)

二、填空题 (本大题共 10 小题，每小题 2 分，共 20 分) 不写解答过程，将正确的答案写在每小题的横线处，错填或不填均无分。

11. 定义类的动态对象数组时，系统只能自动调用该类的\_\_\_\_\_构造函数对其进行初始化。
12. 无论是什么继承方式，派生类的成员不能访问基类\_\_\_\_\_属性的成员。
13. 表达式 `cout<<endl` 还可表示为\_\_\_\_\_
14. 基类的公有成员在派生类中的访问权限由\_\_\_\_\_决定。
15. C++ 支持的两种多态性分别是\_\_\_\_\_多态性和\_\_\_\_\_多态性。
16. C++ 中语句 `const char* const p="hello";` 所定义的指针 `p` 和它所指的内容都不能被\_\_\_\_\_

17. 假定 AB 为一个类，则语句 AB (AB& x); 为该类的\_\_\_\_\_函数的原型说明。

18. 说明常成员函数时，关键字 const 写在成员函数的\_\_\_\_\_和\_\_\_\_\_之间。

19. 在 C++ 中，访问一个对象的成员所用的运算符是\_\_\_\_\_，访问一个指针所指向的对象的成员所用的运算符是\_\_\_\_\_

20. 派生类从一个或多个以前定义的该类的\_\_\_\_\_继承数据和函数。

三、改错题 (本大题共 5 小题，每小题 2 分，共 10 分)

21. 指出下面程序段中的错误，并说明出错原因。

```
class A{
    int a,b;
public:
    A(int aa,int bb){
        a=aa;
        b=bb;
    }
};
A x(2,3),y(4);
```

22. 指出并改正下面利用类模板的对象定义中的错误。

```
template<class T>
class Tany{
    T x,y;
public:
    Tany(T a,T b){
        x=a;
        y=b;
    }
    T sum(){
        return x+y;
    }
};
Tany(int) obj(10,100);
```

23.指出下面程序段中的错误，并说明出错原因。

```
class one{
private:
    int a;
public:
    void func(two&);
};
class two{
private:
    int b;
    friend void one::func(two&);
};
void one::func(two& r){
    a=r.b;
}
```

24.指出下面程序段中的错误，并说明出错原因。

```
#include <iostream>
class A{
public:
    void fun(){
        cout << "a.fun" << endl;
    }
};
class B{
public:
    void fun(){
        cout << "b.fun" << endl;
    }
    void gun(){
        cout << "b.gun" << endl;
    }
};
class C:public A,public B{
private:
```

```

    int b;
public:
    void gun() {
        cout << "c.gun" << endl;
    }
    void hun() {
        fun();
    }
};

```

25. 指出下面程序段中的错误，并说明出错原因。

```

class Location{
    int X,Y=20;
protected:
    int zeroX,zeroY;
    int SetZero(int ZeroX,int ZeroY);
private:
    int length,height;
public:
    float radius;
    void init(int initX,int initY);
    int GetX();
    int GetY();
};

```

四、完成程序题(本大题共 5 小题，每小题 4 分，共 20 分)

根据题目要求，完成程序填空。

26. 在下面横线处填上适当字句，完成类中成员函数的定义。

```

class A{
    int* a;
public:
    A(int aa=0){
        a=_____ ;
    }
    ~A(){_____}
};

```



27. 下面是一个函数模板，用于计算两个向量的和。在下面横线处填上适当字句，完成函数模板定义。

```
#include <iostream>
template<class T>
T* f(T* a,T* b,int n)
{
    T* c=_____ ;
    for(int i=0;i<n;i++)
        c[i]=_____ ;
    return c;
}
void main()
{
    int a[5]={1,2,3,4,5},b[5]={10,20,30,40},*p;
    p=f(a,b,5);
    for(int i=0;i<5;i++)
        cout<<p[i]<<endl;
}
```

28. 下面是一个用户口令检查程序，在横线处填上适当语句完成程序功能。

```
#include <iostream>
#include <string.h>
_____ ;//定义由 PASS 指针所指向的口令 wolleh。

void main()
{
    char user[10]; //用于存放输入的口令
    cout<<"please input your password:"<<endl;
    _____ ;//输入 口令
    if((strcmp(user,PASS))==0)
        cout<<"your password is correct"<<endl;
    else
        cout<<"your password is error"<<endl;
}
```

29.下面是类 fraction(分数)的定义，其中重载的运算符<<以分数形式输出结果，例如将三分之二输出为 2/3。

在横线处填上适当字句，

```
class fraction{
    int den; //分子

    int num; //分母

    friend ostream& operator<<(ostream&,fraction);
    ...
};
ostream& operator<<(ostream& os,fraction fr){
    _____;
    return _____;
}
```

30.在下面程序横线处填上适当的字句，使其输出结果为 0,56,56。

```
#include <iostream>
class base{
public:
    _____ func(){return 0;}
};
class derived:public base{
public:
    int a,b,c;
    _____ setValue(int x,int y,int z){a=x;b=y;c=z;}
    int func(){return (a+b)*c;}
};
void main()
{
    base b;
    derived d;
    cout<<b.func()<<','<<'\n';
    d.setValue(3,5,7);
    cout<<d.func()<<','<<'\n';
}
```

```
base& pb=d;
cout<<pb.func()<<endl;
}
```

五、程序分析题(本大题共 6 小题，每小题 5 分，共 30 分)

给出下面各程序的输出结果。

```
31.#include <iostream>
void main()
{
    int *a;
    int *&p=a;
    int b=10;
    p=&b;
    cout<<*a;
}
```

输出为: \_\_\_\_\_

```
32.#include <iostream>
template<class T>
Tf(T* a,T* b,int n){
    Ts=(T)0;
    for(int i=0;i<n;i++)
        s+=a[i]*b[i];
    return s;
}
void main()
{
    double c[5]={1.1,2.2,3.3,4.4,5.5};
    double d[5]={10.0,100.0,1000.0};
    cout<<f(c,d,5)<<endl;
}
```

输出为: \_\_\_\_\_

## 面向对象程序设计期末 3 参考答案

### 一、单项选择题

1.D 2.B 3.B 4.A 5.C | A 6.A 7.B 8.D 9.A 10.C

### 二、填空题

11. 拷贝

12. private

13. cout << "\n";

14. 继承方式

15. 编译 运行

16. 修改

17. 构造

18. 函数头 函数体

19. . -->

20. 基类

### 三、改错题

21. y(4) 改成 y(4,0)

22. Tany(int) 改成 tany<int>

23. void func(two&) 错误，不能前向申明

24. void hun() {  
    fun();  
}

中的 fun(); 改成 A::fun(); 或 B::fun();

25. y=20 出错，类定义中不能显式地将类的数据成员初始化

### 四、完成程序题

26. new int(aa)

---

```
delete a  
27. new T[n]  
a[i]+b[i]  
28. char* PASS = "wolleh"  
cin >> user;  
29. cout << den << "/" << num << endl;  
os  
30. int  
void
```

## 五、程序分析题

31. 10  
32. 2531

## 面向对象程序设计期末 4

### 一、单项选择题

1. C++对 C 语言有很多改进，下列描述中\_\_\_\_\_使得 C 语言发生了质变，即从面向过程变为面向对象。

- A. 增加了一些新的运算符
- B. 允许重载，并允许设置缺省值
- C. 函数说明必须用原型
- D. 引进了类和对象的概念

2. 设置虚基类的目的为\_\_\_\_\_

- A. 简化程序
- B. 消除二义性
- C. 提高运行效率
- D. 减少目标代码

3. 下面概念不正确的是

- A. 构造函数的名字必须与类名相同
- B. 构造函数、析构函数均无返回值
- C. 析构函数可以被重载
- D. 析构函数可以成为虚拟函数

4. 关于 this 指针不正确的是\_\_\_\_\_

- A. this 指向类对象的首地址
- B. this 不能被对象外部所引用
- C. this 可被该类的所有成员函数所引用
- D. this 不可被友元函数所引用

5. 下列关于运算符重载的正确概念是\_\_\_\_\_

- A. 所有运算符都可以被重载
- B. C++中可定义新的运算符
- C. 重载运算符不能改变原有运算符的参数个数、优先级、结合性
- D. 凡是能被成员函数重载的运算符也一定可被友元函数重载

6. 某类 A 中静态成员函数 display 的原型如下: `static void display();` 则下列说法正确的是\_\_\_\_\_

- A. 该函数可直接使用类 A 的普通数据成员
- B. 该函数可使用 this 指针

C. 该函数只能使用类 A 的静态成员

D. 该函数既可使用类 A 的普通成员函数也可使用类 A 的静态成员函数

7. 下列不属于同一函数原型的是\_\_\_\_

A. `add(int, double)`

B. `int add(int, double)`

C. `int add(int x, double y)`

D. `int add(double y, int x)`

8. 在下列重载函数的调用时可能发生错误的是\_\_\_\_

A. `int print(int x); void print(float);`

B. `int show(int x); int show(char* s);`

C. `int display(Myclass A); char* display(Myclass A);`

D. `int view(int x, int y); int view(int x);`

9. 下列哪个初始化是非法的\_\_\_\_，其中后面选项中的变量类型默认为前面选项出现过的类型。

A. `int i = -1;`

B. `const int ic = i;`

C. `const int* pc = &ic;`

D. `int* const cpic = &ic;`

10. 有如下定义语句：

`int* array_ptr = new int[10];`

`int* pi;`

执行下面哪个语句将使程序执行 `delete[] array_ptr` 时出错\_\_\_\_

A. `(*array_ptr)++;`

B. `array_ptr--;`

C. `pi = array_ptr;`

D. `pi = array_ptr + 1;`

## 二、填空题

1. 已知下列类层次结构，他们都定义了缺省构造函数

`class A {...};`

`class B: public A {...};`

`class C: public B {...};`

`class X {...};`

`class Y {...};`

`class Z: public X, public Y {...};`

`class MI: public C, public Z {...};`

定义 MI mi 的构造函数的执行顺序是\_\_\_\_\_

```
2. #include <iostream>
using namespace std;
int c;
int& funny(int& a, int b)
{
    if (a != b)
        c = a + b;
    a = b;
    b = c;
    return c;
}
int main()
{
    int x = 3, y = 4, z = 5;
    cout << (funny(x, y) = z) << endl;
    cout << "x is " << x << ", and y is " << y << endl;
    z = funny(x, y);
    cout << "x is " << x << ", and y is " << y << ", and z
is " << z << endl;
    return 0;
}
```

程序的运行结果为: \_\_\_\_\_

```
3. #include <iostream>
using namespace std;
class base {
public:
    virtual void func1() {
        cout << "base func1" << endl;
    }
    virtual void func2() {
        cout << "base func2" << endl;
    }
    void func3() {
        cout << "base func3" << endl;
    }
};
class derived :public base {
public:
    void func1() {
```



```

        cout << "derived func1" << endl;
    }
    void func2(int x) {
        cout << "derived func2" << endl;
    }
    void func3() {
        cout << "derived func3" << endl;
    }
};

int main() {
    derived d;
    base& bp = d;
    bp.func1();
    bp.func2();
    bp.func3();
    return 0;
}

```

程序的运行结果为: \_\_\_\_\_

```

4. #include <iostream>
using namespace std;
int num = 10;
class one_class {
    int num;
public:
    void set1(int i) {
        num = i;
    }
    void set2(int& i) {
        int num = 0;
        i = num;
    }
    int get() {
        return num;
    }
    ~one_class() {
        ::num = num;
    }
};

int main() {
    one_class obj;
    obj.set1(100);
    obj.set2(num);
}

```

```

    cout << num << endl;
    cout << obj.get() << endl;
    cout << num << endl;
    return 0;
}

```

程序的运行结果为: \_\_\_\_\_

```

5. #include <iostream>
using namespace std;
class A {
    int i;
public:
    A(int temple) :i(temple) {
        cout << "A constructing\t" << i << endl;
    }
};
class B {
public:
    B() {
        cout << "B constructing." << endl;
    }
};
class C :public A, public B {
public:
    C(int temple) :A(temple), B() {
        cout << "C constructing." << endl;
    }
};
class D :virtual public C {
    int i;
public:
    D(int templ, int tempt) :C(templ), i(tempt) {
        cout << "D constructing." << endl;
    }
};
class E :virtual public C {
public:
    E(int templ) :C(templ) {
        cout << "E constructing." << endl;
    }
};
class F :public D, public E {
    A a;
}

```

```
public:
    F(int i, int j, int k, int l, int m) :E(i), D(j, k),
    C(l), a(m){}
};
int main() {
    F f(1, 2, 3, 4, 5);
    return 0;
}
```

程序的运行结果为: \_\_\_\_\_

### 三、阅读程序 填写代码

```
1. #include <iostream>
#include <string.h>
class string {
    char* str;
public:
    string() {
        str = new char[1];
        *str = '\0';
    }
    string(char* s) {
        if (s) {
            _____
        }
        else {
            str = new char[1];
            *str = '\0';
        }
    }
    string(const string& obj) {
        char* s = "copy from";
        str = new char[strlen(s) + 1];
        strcpy(str, s);
        strcpy(str + strlen(s), obj.str);
    }
    string& operator=(const string& obj) {
        _____
        delete[] str;
        str = new char[strlen(obj.str) + 1];
        _____
        return *this;
    }
}
```

```

    }
    void print() {
        std::cout << str << std::endl;
    }
    ~string() {
        delete[] str;
    }
};
int main() {
    _____
    _____
    string obj3;
    _____
    obj1.print();
    obj2.print();
    obj3.print();
}

```

程序的运行结果为：

```

world
copy from world
world

```

#### 四、找出不能通过编译的语句，并解释原因

```

1  #include <iostream>
2  using namespace std;
3  class base{
4      int base_i;
5  protected:
6      char base_c;
7  public:
8      float base_f;
9      base(int i, char c, float f) :base_i(i),
10     base_c(c), base_f(f) {}
11     void base_set(int i, char c, float f) {
12         base_i = i;
13         base_c = c;
14         base_f = f;
15     }
16 };
17 class derived :protected base {
18     int derived_i;

```

```

19 protected:
20     char derived_c;
21 public:
22     float derived_f;
23     derived(int i, char c, float f) :base(i, c, f),
24     derived_i(i), derived_c(c), derived_f(f) {}
25     void derived_set(int i, char c, float f) {
26         base_i = i;
27         derived_i = i;
28         base_c = c;
29         derived_c = c;
30         base_f = f;
31         derived_f = f;
32     }
33 };
34 struct final:derived {
35     static int count = 0;
36     final(int i, char c, float f):_derived(i, c, f){}
37     void final_set(int i, char c, float f) {
38         base_c = c;
39         derived_c = c;
40         base_f = f;
41         derived_f = f;
42     }
43 };
44 int main() {
45     final fobj;
46     fobj.derived_set(1, 1, 1);
47     fobj.base_set(1, 1, 1);
48     return 0;
49 }

```

## 五、简答题

- 1.友元运算符函数和成员运算符函数有什么不同？
- 2.什么是纯虚函数？什么是抽象类？定义抽象类的作用是什么？

## 六、编程题

- 1.有下列主函数：

```

int main() {
    array<int> array_int(s);

```

```
array<char> array_char;  
array<float> array_float(10);  
}
```

请编写一个能够处理上述简单数据类型的 array 类模板，在该类模板中只需要写出必要的数据成员和构造函数即可；附：写出析构函数。

## 2. 给出类

```
class three_d {  
    int x, y, z;  
public:  
    three_d(int i, int j, int k) :x(i), y(j), z(k) {}  
    three_d() :x(0), y(0), z(0) {}  
};
```

针对给出的类，重载 "\*" 运算符，使 "\*" 运算符可以接受下面的运算形式：

ob\*int, int\*ob; 附：本科生只需完成上面运算的任意一种即可。

## 面向对象程序设计期末 4 参考答案

### 一、单项选择题

1.D    2.B    3.C    4.C    5.C    6.C    7.D    8.C    9.D    10.B

### 二、填空题

1. A B C X Y Z MI

2. 5

x is 4, and y is 4

x is 4, and y is 4, and z is 5

3. derived func1

base func2

base func3

4. 0

100

0

5. A coustructing 4

B coustructing

C coustructing

D coustructing

E coustructing

A coustructing 5

### 三、阅读程序 填写代码

1. str = new char[strlen(s)+1];

2. if(this == &obj) return \*this;

3. return \*this;

4. string obj1("world");

5. string obj2(obj1);

6. obj3 = obj1;

### 四、找出不能通过编译的语句，并解释原因

26 行 base\_i 为私有成员

35 行 静态数据成员不能在类内初始化

45 行 没有默认的构造函数

47 行 base\_set 为私有成员

## 第十二讲补充习题

定义抽象基类 Shape，由它派生出 5 个派生类：Circle（圆形）、Square

（正方形）、Rectangle（矩形）、Trapezoid（梯形）、Triangle（三角形）。用虚函数分别计算几种图形面积，并求它们的和。要求用基类指针数组，使它每一个元素指向一个派生类对象。

## 第十三讲补充习题

文件处理，要求：

- （1）当用户从键盘输入某个文件名后，程序可以将该文件内容全文显示在屏幕上；
- （2）程序可以统计该文件中的总字符数；
- （3）程序可以统计该文件中的单词（以空格为单词分隔符）个数；
- （4）显示上面统计出的两个数字。

## 第十二讲补充习题答案

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

//定义抽象基类 Shape
class Shape
{
public:
    virtual double area() const =0;    //纯虚函数
};

//定义 Circle(圆形) 类
class Circle:public Shape
{
protected:
    double radius;    //半径
```



```
public:
    Circle(double r)
    {
        radius=r;
    }    //构造函数

    virtual double area() const
    {
        return 3.14159*radius*radius;
    };    //定义虚函数
};
```

//定义 Square (正方形) 类

```
class Square:public Shape
{
protected:
    double side;
public:
    Square(double s)
    {
        side=s;
    }    //构造函数

    virtual double area() const
    {
        return side*side;
    }    //定义虚函数
};
```

//定义 Rectangle (矩形) 类

```
class Rectangle:public Shape
{
protected:
    double width,height;    //宽与高
public:
    Rectangle(double w,double h)
    {
        width=w;
        height=h;
    }
};
```

```
    }    //构造函数

    virtual double area() const
    {
        return width*height;
    }    //定义虚函数

};

//定义Trapezoid(梯形)类
class Trapezoid:public Shape
{
protected:

    double top,bottom,height; //上底、下底与高
public:
    Trapezoid(double t,double b,double h)
    {
        top=t;
        bottom=t;
        height=h;
    } //构造函数
    virtual double area() const
    {
        return 0.5*(top+bottom)*height;
    } //定义虚函数
};

//定义Triangle(三角形)类
class Triangle:public Shape
{
protected:

    double width,height; //宽与高
public:
    Triangle(double w,double h)
    {
        width=w;
        height=h;
    } //构造函数
```

```

    virtual double area() const
    {
        return 0.5*width*height;
    } //定义虚函数
};

int main()
{
    Circle circle(12.6); //建立 Circle 类对象 circle

    Square square(3.5);    //建立 Square 类对象 square

    Rectangle rectangle(4.5,8.4); //建立 Rectangle 类对象
rectangle
    Trapezoid trapezoid(2.0,4.5,3.2); //建立 Trapezoid 类对
象 trapezoid

    Triangle triangle(4.5,8.4);    //建立 Triangle 类对象
    Shape
    *pt[5]={&circle,&square,&rectangle,&trapezoid,&triangle};
    //定义基类指针数组 pt，使它每一个元素指向一个派生类对象

    double areas=0.0; //areas 为总面积
    for(int i=0;i<5;i++)
    {
        cout<<pt[i]->area()<<endl;
        areas=areas+pt[i]->area();
    }
    cout<<"total of all areas="<<areas<<endl;    //输出总面积
    return 0;
}

```

### 第十三讲补充习题答案

```

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

```

```
int main()
{
    char ch;
    int letter=0, word=1;
    char filename[30];
    cout<<"Enter a file name:";
    cin>>filename;
    ifstream f1(filename);
    if(!f1)
    {
        cout<<"Cannot open the file."<<endl;
        return 1;
    }
    while(f1)
    {
        f1.get(ch);
        cout<<ch;
        letter++;
        if(ch==' '||ch=='\n')
            word++;
    }
    cout<<endl<<"Total letters: "<<letter;
    cout<<endl<<"Total words: "<<word<<endl;
    return 0;
}
```

## “计算机程序设计”机考试题

总分：100 考试时间：150 分钟

### 1. (20 分)

求 100~999 之间所有各位数字的立方和等于 1099 的整数。

编程要求和评分标准：

- (1) 能正确取出百位、个位数字；(5 分)
- (2) 能正确取出十位数字；(5 分)
- (3) 循环结构，分支结构正确；(5 分)
- (4) 输出格式正确。(5 分)

### 2. (20 分)

定义 5×5 二维数组，找出第 m 行中元素的最大值，并输出最大值元素以及所在的列号。

编程要求和评分标准：

- (5) 数组及变量定义正确 (5 分)
- (6) 键盘输入，并按 5 行 5 列格式输出原数组 (5 分)
- (7) 求最大值及列号功能正确 (5 分)
- (8) 输出格式正确，格式如下： (5 分)

当 5×5 数组为  $\begin{pmatrix} 45 & 87 & 32 & 34 & 54 \\ 54 & 56 & 76 & 78 & 66 \\ 46 & 78 & 87 & 23 & 43 \\ 12 & 32 & 44 & 56 & 78 \\ 55 & 66 & 77 & 88 & 99 \end{pmatrix}$  时，如果用户输入 m=2 (即第 3

行)，则程序输出：第 3 行的最大值：87，所在列号：3

### 3. (20 分)

编写一个函数，用于去掉字符串尾部的空格符，其原型为：

```
char *mytrim(char *string);
```

其中参数 `string` 为字符串，返回值为指向 `string` 的指针。

并编出主函数进行验证。

编程要求和评分标准：

- (1) 子函数头设计正确；(5 分)
- (2) 子函数功能正确；(5 分)
- (3) 主函数调用的方法正确；(5 分)
- (4) 输出结果正确；(5 分)

### 4: (20 分)

定义类 `Point` 及其子类 `Circle`，`Point` 为点类，`Circle` 类为圆类。

`Point` 类的数据成员为：

```
double x, y; //点的 x 和 y 坐标
```

`Circle` 类从 `Point` 类继承得到，其增加数据成员为：

```
double radius; //表示圆的半径
```

`Circle` 类增加的成员函数有计算圆面积的函数、计算圆心到坐标原点距离的函数，其原型分别为：

```
double Area( );  
double Distance( );
```

编程要求和评分标准：

- (1) 给出满足要求的完整的 `Point` 类的定义及其成员函数；(5 分)
- (2) 给出满足要求的完整的 `Circle` 类的定义和它的构造函数；(5 分)
- (3) 正确实现 `Circle` 类的其他成员函数；(5 分)

(4) 编写主函数，从键盘输入一个圆心坐标及半径，计算并屏幕输出圆的面积和圆心到原点距离（5 分）

## 5. (20 分)

编写一个程序，可以读入一个 C++ 语言的源文件，每一行加上行号后保存到另一个后缀为 print.txt 的文件中。

程序输入输出样例：

```
Please input file name: e1305.cpp
1  // Exercise 13_5
2  #include <iostream.h>
.....
27      return 0;
28  }
```

编程要求和评分标准：

- (1) 程序开始执行时提示：“Please input file name:”，能正确输入文件名；（5 分）
- (2) 文件打开和关闭正确；（5 分）
- (3) 能正确读取文件数据；（5 分）
- (4) C++ 语言的源文件每一行加上行号后保存到另一个后缀为 print.txt 的文件中。（5 分）

## “计算机程序设计”机考试题答案

### 1.

```
#include <iostream>
```

```
using namespace std;
int main( )
{
    int n, i, j, k;
    for(n=100; n<=999; n=n+1)
    {
        i = n/100;          // 取出 n 的百位数

        j = (n/10)%10;      // 取数 n 的十位数

        k = n%10;           // 取出 n 的个位数

        if(1099==i*i*i+j*j*j+k*k*k)
            cout << n <<"的各位数字立方和等于 1099, 即 1099="<< i
            <<"^3 + "<< j <<"^3 + "<< k <<"^3"<<endl;
    }

    return 0;
}
```

## 2.

```
#include <iostream>
using namespace std;
int main()
{
    int a[5][5];
    for(int i=0;i<5;i++)
    {
        for(int j=0;j<5;j++)
        {
            cin>>a[i][j];
        }
    }//输入矩阵

    cout<<"当 5*5 的矩阵数组为"<<endl;
    for(int i=0;i<5;i++)
    {
        for(int j=0;j<5;j++)
        {
            cout<<a[i][j]<<" ";
        }
    }
```



```

        cout<<endl;

    } //输出矩阵

    int m,x,k;
    cin>>m;
    x=a[m][0];
    k=0;
    for(int i=0;i<5;i++)
    {
        if(x<a[m][i])
        {
            x=a[m][i];
            k=i;
        }
    }

    cout<<"第"<<m+1<<"行的最大值: "<<x<<"，所在列号: "<<k<<endl;
    return 0;
}

```

### 3.

```

#include<iostream>
using namespace std;
char *mytrim(char *string)
{
    char *ptr=string;
    while(*ptr!=0)
        ptr++;
    do
    {
        ptr--;
    }
    while(*ptr==' ');
    * (++ptr)=0;
    return string;
}
int main()
{
    char str[] = "The art of computer programming    ";
    cout << "截取前的原始字符串是: [" <<str<<']'<<endl;

    cout << "截取空格后的字符串是: [" <<mytrim(str)<<']'<<endl;
}

```

```
    return 0;
}
```

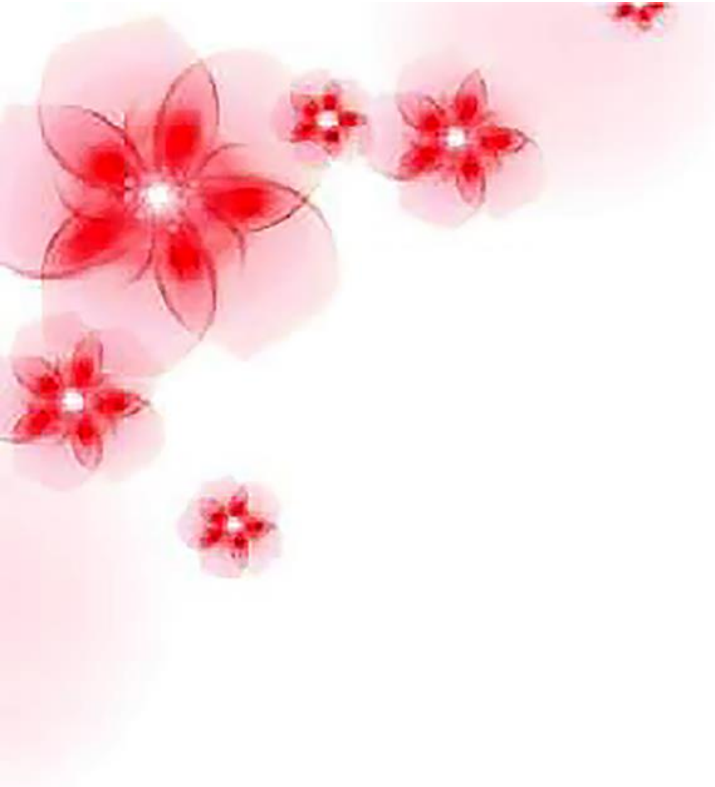
#### 4.

```
#include <iostream>
#include <math.h>
using namespace std;
class Point
{
protected:
    double x,y;
};
class Circle:public Point
{
    double radius;
public:
    Circle(double a=0,double b=0,double c=0)
    {
        x=a;
        y=b;
        radius=c;
    }
    double distance( )
    {
        return sqrt(x*x+y*y);
    }
    double area( )
    {
        return 3.1415926*radius* radius;
    }
};
int main ( )
{
    double a,b,c;
    cin>>a>>b>>c;
    Circle x(a,b,c);
    cout<<x.area( )<<endl;
    cout<<x.distance( )<<endl;
    return 0;
}
```

#### 5.

```
#include <iostream>
```

```
#include <fstream>
#include <iomanip>
using namespace std;
int main ( )
{
    char ch, name[20];
    int line = 1;
    cout<<"Please input file name: ";
    cin>>name;
    ifstream filein(name);
    ofstream fileout("print.txt");
    if(!filein||!fileout)
    {
        cout << "Cannot open file!";
        return 1;
    }
    cout << setw(4) << line << " ";
    fileout << setw(4) << line << " ";
    while(filein)
    {
        filein.get (ch);
        cout << ch;
        fileout << ch;
        if(ch=='\n')
        {
            line++;
            cout << setw(4) << line << " ";
            fileout << setw(4) << line << " ";
        }
    }
    filein.close( );
    fileout.close( );
    return 0;
}
```



更多精彩，尽在南洋书院学生会微信公众号的南卷汇专栏，欢迎通过公众号提供题目或反馈错题信息，南卷汇需要您的支持。

