



## “C++程序设计”考试试卷 (A 卷)

考试方式 开卷 考试日期 2022-11-26 考试时长 150 分钟

专业班级                      学      号                      姓      名

题号	一	二	三	四	五	六	总分	核对人
分值	15	20	20	15	15	15	100	马光志
得分								

分 数	
评卷人	

一、单选题：请从 4 个选项选择一个最合适的选项作为答案（15 分：每小题 3 分）。

解答内容不得超过装订线

1. 关于定义 “union A { int x; const int y = 3; } a = { 5 };”, 如下叙述哪个\_\_B\_\_正确:  
A. a.x=3, a.y=3  
B. a.x=5, a.y=5  
C. a.x=3, a.y=5  
D. a.x=5, a.y=3
2. 关于 “void f(int&) {}; void f(const int&) {}; void f(int&&) {}; int x=2;”, 如下叙述\_\_D\_\_正确:  
A. f(x)调用 void f(int&), f(2)调用 void f(const int&)  
B. f(x)调用 void f(const int&), f(2)调用 void f(int&&)  
C. f(x)调用 void f(int&&), f(2)调用 void f(int&)  
D. f(x)调用 void f(int&), f(2)调用 void f(int&&)
3. 对于说明 “int &f(); int \*&g();” 及其函数调用 f()和 g(), 如下叙述\_\_C\_\_正确:  
A. 调用 f()和调用 g()均不可被赋值  
B. 调用 f()不可被赋值, 调用 g()可被赋值  
C. 调用 f()和调用 g()均可被赋值  
D. 调用 f()可被赋值, 调用 g()不可被赋值
4. 对于定义 “struct A { void f() } a, \*p; struct B: A { virtual void f() } b, \*q=&b;”, 若 p=q 如下叙述\_\_B\_\_正确:  
A. p->f()调用 A::f(), q->f()调用 A::f()  
B. p->f()调用 A::f(), q->f()调用 B::f()  
C. p->f()调用 B::f(), q->f()调用 B::f()  
D. p->f()调用 B::f(), q->f()调用 A::f()
5. 对于定义 “struct A { const int x = 3; volatile int y = 4; } const a;”, 如下叙述\_\_B\_\_正确:  
A. a.x 和 a.y 的类型均为 const int;  
B. a.x 类型为 const int, a.y 的为 const volatile int;  
C. a.x 和 a.y 的类型均为 const volatile int;  
D. a.x 类型为 const volatile int, a.y 的为 const int;

分 数	
评卷人	

二、在最多使用单重作用域例如 A::x 的前提下，在空白处填写以下各类可被访问的成员及其访问权限（20 分：根据正确回答的成员个数按比例计算给分）。

```
class A {           //类 A 的可访问成员：
    int a;          //私有成员： int a(A::a);
protected:        //保护成员： int b(A::b), c(A::c);
```

```

        int b, c;          //公有成员: int d(A::d), e(A::e);_____
public:
        int d, e;
};
class B: protected A { //类 B 的可访问成员:
        int a;          //私有成员: int a(B::a);_____
protected:            //私有成员: _____
        int b, f;       //保护成员: int A::b, b(B::b), c(A::c, B::c), d(A::d, B::d), A::e, f(B::f);____
        using A::d;     //保护成员: _____
public:                //公有成员: int e(B::e), g(B::g)_____
        int e, g;       //公有成员: _____
};
struct C: A {          //类 C 的可访问成员:
        int a;          //私有成员: _____
protected:            //私有成员: _____
        int b, c;       //保护成员: int A::b, b(C::b), A::c, c(C::c), _____
public:                //保护成员: _____
        int e, g;       //公有成员: int a(C::a), d(A::d, C::d), A::e, e(C::e), g(C::g)_____
        using A::d;     //公有成员: _____
};
struct D: B, C {       //类 D 的可访问成员:
        int a;          //私有成员: _____
protected:            //私有成员: _____
        int b, c;       //保护成员: int B::b, C::b, b(D::b), B::c, C::c, c(D::c), B::d, B::f_____
public:                //保护成员: _____
        int e, f;       //公有成员: int C::a, a(D::a), B::e, C::d, C::e, e(D::e), f(D::f), B::g, C::g____
};                      //公有成员: _____

```

分 数	
评卷人	

三、回答 main 中每行语句的输出结果（20 分：前四个语句的输出每个 3 分，后两个语句的输出每个 4 分）。

```

#include <iostream>
using namespace std;
struct A { A() { cout << 'A'; } };
struct B { A a; B() { cout << 'B'; } };
struct C : virtual A { C() { cout << 'C'; } };

```

```

struct D : B, virtual C { D() { cout << 'D'; } };
struct E : virtual A, virtual D {
    D d;
    E() : A() { cout << 'E'; }
};
struct F : B, virtual C, virtual D, E {
    D d; E e;
    F() { cout << 'F'; }
};
int main() {
    A a; cout << '\n';           //输出=A
    B b; cout << '\n';           //输出=AB
    C c; cout << '\n';           //输出=AC
    D d; cout << '\n';           //输出=ACABD
    E e; cout << '\n';           //输出=ACABDACABDE
    F f; cout << '\n';           //输出=ACABDABACABDEACABDACABDACABDEF
}

```

解答内容不得超过装订线

分 数	
评卷人	

四、综合分析并指出以下程序中下划线位置可能出现的语法错误及其原因 (共 15 分：每错约 1 分)。

```

class A {
    int a;
public:
    const int b;
    int c;
    virtual A(*g)() = 0;           //错误 1: 数据成员 g 不能用 virtual 说明_____
    A() { a=b=c=0; }              //错误 2: ()后没对 b 初始化_错误 3: 函数体中不能初始化 b_
    virtual A(int x): a(x),b(x),c(x){ }; //错误 4: 构造函数不能定义为 virtual_____
} a = (4, 3);

class B: A {
    using A::a;                    //错误 5: 不能 using 私有的 A::a_____
    ~B(){}
public:
    int d;
    friend int operator[ ](int) { return 2; };//错误 6: [ ]只能重载为实例函数成员: friend 说明非成员
    B(int x, int y) { d = x + y; };
} b(5, 6);                         //错误 7: b 无法调用私有的析构函数_____

class C: B {

```

```

namespace D{ int z; };          //错误 8: 不能在类体中定义名字空间_____

public:

    ~C(int x) { };              //错误 9: 析构函数不能有参数 _____

    friend void main();

} c;                            //错误 10: 类 B 未提供无参构造函数,无法生成 C::C()_____

template<typename T>  const  T   x = 3;

void  main(){

    int   i = a.a;              //错误 11: main 不能访问私有成员 a.a_____

    int   A::* p = static_cast<int A::*>(&A::b); //错误 12: 不能去掉&A::b 的 const 属性_____

    char &y = x<char>;          //错误 13: 只读变量 x<char>不能被可写引用变量 y 引用_____

    i = b.b + c.d;              //错误 14: main 不能访问私用成员 b.b_____

    i = b.*p;                   //错误 15: A 非 B 的父类且 main 不是 B 的友元, 不能访问 b.*p

}

```

分 数	
评卷人	

五、请填入自己学号的最后一位十进制数字, 计算 **main** 函数中变量 **i** 在每条赋值语句执行后的值 (共 15 分: 每小题 2.5 分)。

```

int   x = 填写自己学号最后一位十进制数_____, y = x + 3;

struct A {
    int   x = ::x + 2;
    static int& y;
public:
    operator int( )const { return x + y; }
    int& v(int& x) {
        for (int y = 1; x < 301; x ^= y, y++)
            if (x > 300) { x -= 31; y -= 2; }
        return ++x;
    }
    A& operator++( ) { ++x; ++y; return *this; }
    A(int x, int y = ::y + 3) { A::y = y; }
};

int& A::y = ::y;

void main( ) {
    A   a(2, 7), b(5);
    int   i, &j = i, A::*p = &A::x;
    auto f = [&i](int x) mutable {++i; return x + i; };
    i = a.y;                      //i=
    j = a.x;                      //i=
    i = a.*p;                     //i=
    i = ++a;                      //i=
    i = b.y + f(1);               //i=
}

```

```

        (b.v(i)=2) += f(3);
    }
    //i=

```

学号	i = a.y	j = a.x	i = a.*p	i = ++a	i = b.y + f(1)	(b.v(i)=2) += f(3)
0	10	2	2	14	27	33
1	10	3	3	15	28	34
2	10	4	4	16	29	35
3	10	5	5	17	30	36
4	10	6	6	18	31	37
5	10	7	7	19	32	38
6	10	8	8	20	33	39
7	10	9	9	21	34	40
8	10	10	10	22	35	41
9	10	11	11	23	36	42

解答内容不得超过装订线

分 数	
评卷人	

六、对于一元多次函数 $f(x) = \sum_{i=0}^{n-1} a_i x^i$ ，其中 $x^i$ 和 $a_i$ 皆为 int 类型的整数，注意  $n \geq 0$ ,  $i$  为  $x$  的指数。现定义函数类 F 如下，试对其中的每个实例函数成员编程(共 15 分：每个函数 1.5 分)。

```

class F {
    int* const a;           //用于存放函数的多项式系数 $a_i$ ， $a_0$ 存放于 a[0]
    const int n;           //多项式的系数个数

public:
    F(int n, ...);          //创建 n 个系数的函数，n<0 抛出异常，...为 n 个系数
    F(const F& f);          //根据已知函数 f 深拷贝构造新函数
    F(F&& f) noexcept;     //根据已知函数 f 移动构造新函数
    F& operator=(const F& f); //深拷贝赋值运算符的重载
    F& operator=(F&& f) noexcept; //移动赋值运算符的重载
    F operator+(const F& f) const; //两个函数相加
    operator int() const noexcept; //获得函数多项式的系数个数 n
    int& operator[](int i); //获得第 i 个系数(i 从 0 开始)，不存在则抛出异常
    int operator()(int x) const; //计算实参为 x 时的函数值
    ~F() noexcept;         //析构函数
};

```

答：

```

F::F(int n, ...): a(n<0?throw "n<0!": (n==0?nullptr: new int[n])), n(a ? n : 0) {

    if (n>0 && a == nullptr) throw "memory not enough!";

```

```

    int* p = &n + 1;

    for (int x = 0; x < n; x++) a[x] = p[x];
}

F::F(const F& f) : a(f.n>0?new int[f.n]:nullptr), n(a ? f.n : 0) {
    if (n > 0 && a == nullptr) throw "memory not enough!";
    for (int x = 0; x < n; x++) a[x] = f.a[x];
}

F::F(F&& f)noexcept : a(f.a), n(f.n) {
    (int*&)f.a = nullptr;    //或*(int**)&f.a=nullptr;
    (int&)f.n = 0;           /*(int*)&f.n = 0;
}

F& F::operator=(const F& f) {
    if (this == &f) return *this;
    if (a) delete a; //或 delete []a;
    if (f.n == 0) {
        (int*&)a = nullptr;
        (int&)n = 0;
        return *this;
    }
    (int*&)a = new int[(int&)n=f.n];
    if (a == nullptr) throw "memory not enough!";
    for (int x = 0; x < n; x++) a[x] = f.a[x];
    return *this;
}

F& F::operator=(F&& f)noexcept {
    if (this == &f) return *this;
    if (a) delete a;    //或 delete []a;
    (int*&)a = f.a;      //或*(int**)&a=f.a;
    (int&)n = f.n;       /*(int*)&n = f.n;
    (int*&)f.a = nullptr; //或*(int**)&f.a=nullptr;

```

```

        (int&)f.n = 0;          /*(int*)&f.n = 0;

        return *this;
    }

    F F::operator+(const F& f)const {
        F r(n > f.n ? *this: f);
        if (n > f.n)
            for (int x = 0; x < f.n; x++) r.a[x] += f.a[x];
        else
            for (int x = 0; x < n; x++) r.a[x] += a[x];
        return r;
    }

    F::operator int()const noexcept { return n; }

    int& F::operator[ ](int i) {
        if (i < 0 || i >= n) throw "subscription error!";
        return a[i];
    }

    int F::operator()(int x)const {
        if (n == 0) throw "can not call a null function!";
        int s = a[n-1];
        for (int k = n - 1; k >= 1; k--) s = s * x + a[k - 1];
        return s;
    }

    F::~F()noexcept {
        if (a!=nullptr) {
            delete a; //或 delete []a;

            (int*&)a = nullptr;
            (int&)n = 0;
        }
    }

```