



华中科技大学计算机科学与技术学院

2023~2024 学年第一学期

“C++程序设计”考试试卷(A 卷)

考试方式: 开卷 考试日期: 2023-12-2 考试时长: 150 分钟

专业班级: 学 号: 姓 名:

题号	一	二	三	四	五	六	七	总分
分数								

分 数		一、单项选择题(每小题 2 分, 共 10 分) 答题卡见右, 以答题卡为准。	1	2	3	4	5
评卷人							

1. 对于如下程序:

```
#include <stdio.h>
struct A{
    const int x;
    volatile int y;
    A(int x, int y):x(x), y(y){ }
};
const A a(2,3);
```

则 a.x 和 a.y 的类型分别为_____:

- A. const int, volatile int B. const int, const volatile int
C. const int, const int D. const volatile int, volatile int

2. 对于如下程序:

```
#include <stdlib.h>
struct A{ A( ) { }; ~A()=default; }a;
int main( ){ A b; exit(0); }
```

则关于对象 a, b 的析构, 如下哪个叙述正确_____:

- A. 没有析构函数, 故都没有析构; B. a 析构了但是 b 没析构;
C. b 析构了但是 a 没析构; D. 都析构了;

3. 对于定义 “const char *&g();”, 如下哪个语句是错误的_____:

- A. g()="abcde"; B. *g()='A';
C.const char *p=g(); D. const char*&const(&q)()=g;

4. 对于如下类型定义:

```
struct A{ virtual void f( ) { } }a, &b=a, &&c=A( );
```

欲调用函数 void g(A&&x), 以下哪个表达式做实参没有语法错误_____:

- A. a B. b
C. c D. A()

5. 关于类 E 的友元函数的叙述哪个最合适_____:

- A. 能够访问 E 的私有成员 B. 能够访问 E 的保护成员
C. 能够访问 E 的公开成员 D. 能够访问 E 的所有成员

分 数		二、对错判断题(每小题 2 分, 共 10 分) 答题卡见右, 以答题卡为准。请只答“对”或“错”。				
评卷人		1	2	3	4	5

1. 同一个类中可以同时定义函数成员 static void g()和 virtual void g(): _____。

2. 对于定义 const char*const&p="abc"; 这条语句编译正确: _____。

3. 对于全局变量 m 的定义 struct A{ int x=2; public: A(int a): x(x+a){ } }m(4); 则有 m.x=6: _____。

4. 将主函数 main 定义为一个类的友元时, 不能同时在这个类的体内定义 main 的函数体: _____。

5. final 函数必须定义为虚函数: _____。

分 数		三、假定最多只允许一级作用域如 A::x 访问, 指出以下各类可访问的实例数据成员及其访问权限(20)。	
评卷人			

```
class A {
    int a, h, k;
protected:
    int b, e;
public:
    int c, d;
```



```
};
class B: public A {
    int d;
protected:
    int c, e;
public:
    int b, f;
};
class C: protected A {
    int g;
protected:
    int h, d;
public:
    int b, i;
};
struct D: protected B, public C {
    int j;
protected:
    int k;
public:
    using B::c;
    int d, n, p;
};
```

分 数	
评卷人	

四、指出 **main** 中每行的运行结果 (第 1 小题 6 分, 第 2 小题 14 分, 共 20 分)。

1. 试给出 **main** 函数中每行变量 **c** 的值 (每行 1 分)。

```
struct A {
    virtual char f() { return 'A'; }
}a,*p=&a;
struct B: A { char f() { A::f(); return 'B'; }}b;
void main() {
    char c= a.A::f(); //c=
    c = b.f(); //c=
    c = b.A::f(); //c=
    c=p->f(); //c=
    p = &b; c = p->A::f(); //c=
    c = p->f(); //c=
}
```

2. 试给出 **main** 函数中每行的输出结果 (每行分值依次为: 1, 1, 3, 4, 5)。

```
#include <iostream>
using namespace std;
struct A { A() { cout << 'A'; } };
struct B { B() { cout << 'B'; } };
struct C : virtual A { C() { cout << 'C'; } };
struct D : B, C { D() { cout << 'D'; } };
struct E : virtual A, virtual D {
    D d;
    E(): A() { cout << 'E'; }
};
void main() {
    A a; cout << '\n'; //输出=
    B b; cout << '\n'; //输出=
    C c; cout << '\n'; //输出=
    D d; cout << '\n'; //输出=
    E e; cout << '\n'; //输出=
}
```

分 数	
评卷人	

五、指出以下程序下划线处的语法错误及其原因 (每错约 1 分, 共 10 分)。

```
class A {
    int a;
protected:
    ~A() {}
public:
    const int &b;
    int c;
    virtual A(*g)(int, int)=0;
    A(int x){_a = x; };
} x;
class B: A {
    int b, d;
    using A::a;
public:
    B(int x, int y, int z){_d = x + y + z;};
```



```

} b(5, 6, 7);
int main() {
    int A::*p = &x.A::b;
    int i = b.b;
    i = i + b.d;
    return b.*p;
}

```

分 数	
评卷人	

六、请填入本人学号的最后一位十进制数字，指出 **main** 函数中变量 **i** 在每条赋值语句执行后的值 (每小题 2.5 分，共 15 分)。

```

int x=学号最后一位十进制数_____, y=x+5;
struct A {
    int x;
    static int y;
public:
    operator int() const volatile { 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:::x+1, int y:::y+3) { A::x=x; A::y=y; }
};
int A::y:::x;
void main() {
    A a(2, 7), b(5), c;
    int i, &j=i, A::*p=&A::x;
    i=a.y; //i=
    j=a.x; //i=
    i=a.*p; //i=
    i=b.y+::y; //i=
    i=++a; //i=
    (b.v(i)=3)+=x; //i=
}

```

分 数	
评卷人	

七、编程题:有一个剧场共有 500 个座位，座位编号为 1~500。在剧场的入口有一个刷票机，用于记录每次演出的入场座位编号，假定买了票的观众都会入场，刷票机刷票并记录每个座位编号，刷票机用座位编号 0 记录入场结束，或本次入场记录的座位编号序列结束。请定义一个最多能记录 **m** 场演出的入场登记表类 **REG**，用于登记刷票机每场演出的入场记录座位编号序列。(每 1 个函数成员 1.5 分，共 15 分)。

```

class REG {
    int **const e; //e[k]用于记录第 k 场演出按序入场的座位编号
    const int m; //能够记录的最大演出次数 m
    int c; //已经记录的演出次数
public:
    REG() noexcept; //用于构造空登记表
    REG(int m); //用于构造最多记录 m 次演出的登记表
    REG(const REG&); //登记表深拷贝构造函数
    REG(REG&&) noexcept; //登记表移动构造函数
    REG& operator=(const REG&); //登记表深拷贝赋值运算符重载函数
    REG& operator=(REG&&) noexcept; //登记表移动赋值运算符重载函数
    REG& operator<<(const int*); //将一次演出的入场座位编号序列存入登记表
    bool sold(int k, int s); //查询第 k 场(k 可为 0)的座位号 s 是否售出，售出返回真，否则返回假
    int* operator[](int x); //返回演出序号为 x 的按序入场座位编号序列
    ~REG() noexcept; //登记表析构函数
};

```