

2006 级《C++面向对象程序设计》期末考试试题 (A 卷)

考试时间: 2008 年 1 月 15 日

班级 _____ 学号 _____ 姓名 _____

- ✧ 本试卷满分 100 分;
- ✧ 请将答案写在答题纸上, 写明题号, 不必抄题, 字迹工整、清晰;
- ✧ 请在答题纸和试题纸上都写上你的班级, 学号和姓名, 交卷时请将试题纸、答题纸和草纸一并交上来。

一、单选题(共 20 分, 每题 2 分)

1. 已知函数 `int f();` 是类 A 的公有静态成员, a 是用 `const A a;` 的形式创建的一个对象, 那么下列语句错误的是:
(A) `int num=A::f();` (B) `int num=a.f();`
(C) `int num= a.f() const;` (D) `const int num= a.f();`
2. 在实现函数 `const A & f(A & obj)` 时, 下面的哪一个表达式不能作为函数的返回值?
(A) 全局静态变量 (B) 局部静态变量
(C) 函数参数 obj 对象 (D) 局部自动变量 (auto 变量)
3. 下面关于 new 和 delete 操作符的说法, 哪个是不正确的:
(A) 使用 new 操作符, 可以动态分配全局堆中的内存资源。
(B) 用 new[] 申请的数组, 应该用 delete[] 释放。
(C) 执行语句 `A * p = new A` 成功后, 调用了 `delete p;`, 则 p 的值为 0。
(D) 执行语句 `A * p = new A[10];` 时, 会调用 10 次构造函数。
4. 下列哪种成员函数可以由 const 修饰?
(A) 无参构造函数 (B) 拷贝构造函数 (C) 自定义的转换函数 (D) 虚的析构函数
5. 类 D 以 public 方式继承了类 B, 如下的哪个声明不能作为类 D 的成员?
(A) `B* pB;` (B) `B aB;` (C) `D aD;` (D) `D* pD;`
6. 若类 C 的定义为 `class C:public A,private B{ public: int one,two; }`, 则说法正确的是:
(A) one 和 two 的空间分配及初始化的顺序, 要根据类 C 的构造函数的初始化列表。
(B) A 和 B 的构造顺序, 要根据类 C 的构造函数的初始化列表。
(C) 若 A、B、C 中均无虚函数且都有非静态数据成员, 则 `sizeof(C)=sizeof(A)+sizeof(B)+2*sizeof(int)`。
(D) 类 A 和类 B 不能有公共的基类。
7. 可以在类 A 的成员函数 `void f() const` 的函数体中使用的 this 指针类型是:
(A) `A*` (B) `const A*` (C) `const A * const` (D) `A * const`
8. 判断类 A 的两个对象 a1 与 a2 是否是同一个对象的方法是:
(A) 利用调试器, 查看 a1 与 a2 各成员数据的值是否相等
(B) 重载 == 运算符, 用它来判断各成员数据的值是否相等。
(C) 判断这两个对象的地址值是否相等。
(D) 比较创建时使用的构造函数的实参是否相同。

9. 下面表达式中不可能出现的是:

- (A) a['a'] (B) f(100)=0 (C) ~A(0) (D) P<int> (0)

10. 下面关于纯虚函数的说法, 哪种是错误的?

- (A) 纯虚函数一定是类的非静态成员函数。
(B) 纯虚函数可以是带 `const` 修饰的常量成员函数。
(C) 含有纯虚函数的类一定是抽象类。
(D) 纯虚函数的返回值类型一定是 `void`。

二、判断正误, 对于你认为错误的论述, 说明原因或举出反例。(共 20 分, 每题 2 分)

1. 类的所有成员函数都可以访问类中静态成员函数, 但静态成员函数不能访问类的非常量成员函数, 只能访问类的常量成员函数 (用 `const` 修饰的成员函数)。
2. 在同一个类中, 可以定义重载的成员函数 `int f(int anInt=100);` 和 `void f(int anInt);`。
3. 私有继承中, 基类中所有成员对派生类来说都是不可见的。
4. 只要一个类存在一个公有的无参的构造函数, 就一定可以用这个类创建对象。
5. 类 D 以 `public` 方式继承类 B, 若在这两个类中以同样的方式分别声明了一个同名的整型成员变量 `x`, 那么, 为类 D 的对象分配内存空间时, 不需要为类 B 中声明的 `x` 分配空间。
6. 在任何情况下, 都不可能通过构造函数的初始化列表对类的静态成员变量进行初始化。
7. 使用形如 `virtual class A: public class B { }` 的语法可以声明一个虚基类。
8. 一个函数在基类中没有声明为虚函数, 但在派生类中被重写时被声明为虚函数, 那么基类中的这个函数也会被看作是虚函数。
9. 类中的非静态成员函数越多, 则该类产生的对象所占内存空间就越大。
10. 若不考虑继承, 则 `protected` 访问控制等同于 `private`。

三、指出下列程序代码片断中存在的不合理之处, 并简要说明理由。(共 10 分, 每题 2 分)

1. <code>A* pa; pa->f();</code>	2. <code>A* pa=new A[10]; delete pa;</code>	3. <code>A obj(); obj.f();</code>
4. <code>class A { const int a=0, b=0; };</code>	5. <code>A& f(){ return A(); }</code>	

四、回答下列各题 (每题 4 分, 共 20 分)

1. 可以用 `class A;` 的语法形式声明一个类, 该语法中并没有指出类的具体内容。请举例说明这种声明形式在编程时有何用处?
2. 说明 `A* const p = new A;` 和 `A* p=new A;` 在含义上有什么不同?
3. 拷贝构造函数的使用场合有哪些? 至少列举 3 种不同情况。
4. 如何定义类 A, 使得它的对象 `a` 可以作为函数 `void f(int)` 的实参。
5. 定义并实现一个类时, 在哪些情况下必须使用初始化列表? 至少列举 3 种情况。

五、阅读下面两个类的定义和部分实现代码，完成 4 个问题。（共 12 分，每小题 3 分）

<pre>#include<iostream.h> class B { public: static int count; B(int v) : nB(v) { ++count; } ~B() { --count; } B * Clone(); //--- (问题 1) void Display() { cout<<count <<endl; } private: B(const B& rhs):nB(rhs.nB) { ++count; } int nB; };</pre>	<pre>int B::count = 0; void main() { A a(3); a[0]-> Display(); B* p1 = a[1]->Clone(); a[1]->Display(); B* p2 = a[2]->Clone(); a[2]->Display(); delete p2; delete p1; a[0]->Display(); }</pre>
<pre>class A { public: A(int n) { for(int i=0;i<5;++i) bs[i]= new B(n); } ~A() { for(int i=0;i<5;++i) delete bs[i]; } A& operator=(const A& rhs); //---(问题 4) const B * operator[](int idx) const { return bs[idx]; } //----(问题 2) B * operator[](int idx) { return bs[idx]; } //----(问题 2) private: B * bs[5]; };</pre>	

1. 代码中 Clone 函数的含义是：在全局堆中，根据当前对象，创建一个一致的新对象（当前对象的一个拷贝），并返回新对象的地址。请实现 Clone 函数，使得整个程序运行时输出的四个数据依次为 5，6，7，5。
2. 类 A 中定义并实现了 operator[]操作，从而可以对 A 类的对象进行下标运算。请根据你对 const 修饰符的理解，说明两个重载的 operator[]在使用场合方面有什么不同？
3. 若考虑将来可能从 B 类派生新的子类，类 B 应该修改或增加哪些成员的定义，从而使类 B 的设计更合理？对于每个变动，说明理由。
4. 请写出类 A 的重载赋值操作的实现代码，该实现能够完成 A 类对象的深赋值。

六、分别写出下面两个程序的运行结果（共 8 分，每题 4 分）

1.

<pre>#include <iostream.h> class A { public: A(int n):val(n){} protected: int val; };</pre>	<pre>class B : public A { public: B(int n):A(n) { pB = (n>0 ? new B(n-1) : 0); } ~B() { delete pB; } void Display() { cout<<val<<endl; if (pB!=0) pB->Display(); } private: B * pB; };</pre>
<pre>void main() { B b(4); b.Display(); }</pre>	

2.

<pre>#include <iostream.h> class A { public: virtual ~A() {} void f() {g();h();} virtual void h() {cout<<"A::h()" <<endl;} protected: void g() {cout<<"A::g()" <<endl;} };</pre>	<pre>class C : public A, private B { public: C(int n1,int n2) : num(n1), B(n2) { } virtual void h() { cout<<"num=" <<num<<endl; cout<<"val=" <<val<<endl; } protected: void g() {cout<<"C::g()" <<endl;} private: int num; };</pre>
<pre>class B { public: B(int n):val(n){ } protected: int val; };</pre>	<pre>void main() { C c(1,2); c.f(); }</pre>

七、某程序员在完成一个软件开发任务过程中，需要实现并使用许多函数，他把这些函数分别封装到一些类中，这样的类称作命令类。这些函数在设计时需要考虑可能会现的各种异常，由于对这些异常的处理方法存在共性，所以他不愿在每个命令类的实现过程中分别处理这些异常，而打算只在一个地方处理一次，于是，他为所有的命令类设计了一个基类 **Command**，请根据他对该基类的定义及部分实现，分析此程序员的设计意图并回答问题。（共 10 分）

```
class Command {
public:
    bool Execute() {
        try {
            Run();
            return true;
        }
        catch(...) {
            return false;
        }
    }
protected:
    // Run 函数的定义，对应下面的问题 1 .
};
```

1. **Run** 是 **Command** 类的一个具有 **prtotedced** 访问权限的成员函数，请给出它的定义（函数原型）。（3 分）
2. 假设某个命令类 **HelloCommand** 的对象能够输出 "Hello"，请在 **Command** 类的基础上，利用继承机制，完成此命令类的定义和实现。（4 分）
3. 编写 **main** 函数，其功能是：产生 **HelloCommad** 类的一个对象，并让该对象输出 "Hello"（3 分）

（全卷完）