

📑 题目列表 (/course/46/contest/608/home)

📊 提交状态 (/course/46/contest/608/submissions)

🏆 排行榜 (/course/46/contest/608/ranklist/normal)

单选题

刷新 ↺

单选题

说明:

- 每小题分值相同
- 考试期间的该题的任意提交都被判定为100分，考试结束后将进行重测。

1. 下面说法错误的是：

- A. C++11中，类内的非静态成员变量可以在类的构造函数外指明初始化参数。
- B. 常量对象可以成为非常量成员函数的参数
- C. 静态成员函数可以访问 this 指针。
- D. 对于自定义类型的静态数据成员，如果在.h文件中同时完成声明和定义，则链接可能会失败
- E. 创建和删除对象时，new[]和delete同时使用可能会导致内存泄漏

2. 关于下列代码的描述不正确的是（编译时添加选项--std=c++11 -fno-elide-constructors）

```

#include <iostream>
using namespace std;

class MyInt{
public:
    MyInt(int i = 0) : val(i)  // (a)
    {
    }
    MyInt(const MyInt& src) : val(src.val)  // (b)
    {
    }
    MyInt(MyInt&& src) : val(src.val)  // (c)
    {
    }
    operator int() const  // (d)
    {
        return val;
    }
    MyInt& operator=(MyInt&& rhs)  // (e)
    {
        val = rhs.val;
        return *this;
    }
    MyInt& operator=(const MyInt& rhs)  // (f)
    {
        val = rhs.val;
        return *this;
    }
    int get_value() const
    {
        return val;
    }
private:
    int val;
};

void print(MyInt obj)
{
    cout << obj.get_value() << endl;
}

void print(int value)
{
    cout << value << endl;
}

int main()
{
    MyInt a;
    a = -3.8;  // (1)
    auto b = a + 1;  // (2)
    auto c = a;  // (3)
    print(a); print(b); print(c);  // (4)
    return 0;
}

```

- A. (1)处会先将double转换为int，再调用(a)构造对象，最后调用(e)进行移动赋值
- B. (2)处会先调用(d)将变量a转换为int，执行加法，然后调用(a)构造对象，最后调用(c)进行移动构造
- C. (3)处会调用(b)进行拷贝构造

D. (4)处三个语句的输出是 -3\n-2\n-3\n，其中\n代表换行符

3. 关于以下代码，说法正确的是

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

class Base {
public:
    virtual void f1(float x) {
        cout << "Base::f1 " << x << endl;
    }
    void f2(float x) {
        cout << "Base::f2 " << x << endl;
    }
};

class Derived : public Base {
public:
    virtual void f1(float x) {
        cout << "Derived::f1 " << x << endl;
    }
    void f2(int x) {                                     // (1)
        cout << "Derived::f2 " << x << endl;
    }
};

int main(void) {
    Base x;
    Derived y;
    Base* pb1 = &x;
    Base* pb2 = &y;
    Derived* pd = &y;

    pd->f2(3.14f);                                       // (2)

    return 0;
}
```

A. Derived 类不仅继承了 Base::f2(float x) ，而且额外定义了函数名相同，函数参数不同的 Derived::f2(int x) ，从而实现了函数重载。

B. Derived::f1() 实现了对 Base::f1() 的重写覆盖。

C. (1) 处发生了重写隐藏。如果在此处加入 override 关键字变为 void f2() override (int x) ，则可以实现重写覆盖。

D. 由于 (1) 处发生了重写隐藏，因此 (2) 处将输出 Derived::f2 3.14 。

4. 下面关于构造函数和析构函数，下列说法错误的是：

A. 在一个类中可以有多多个构造函数，它们构成了函数的重载；析构函数没有参数，所以不能重载，一个类仅有一个析构函数。

B.

```
#include<iostream>
using namespace std;
class P
{
public:
    P(){cout << "调用构造函数" << endl;}
    ~P(){cout << "析构函数的调用" << endl;}
};
void test(){P p;}
int main()
{
    test();
    P p1;
    return 0;
}
```

运行上述代码，输出为

```
调用构造函数
调用构造函数
析构函数的调用
析构函数的调用
```

C. 通常，拷贝构造函数的形参被设定为本类对象的常量引用

D. 构造函数没有返回值类型，函数名与类名相同

5. 下面关于设计模式，错误的是：

A. 结构型模式关注对象之间结构关系上的抽象，从而提升对象结构的可维护性、代码的健壮性，能在结构层面上尽可能的解耦合

B. 如果在一个系统里面有许多类，它们仅仅在行为上有区别，那么使用策略模式可以动态地让一个对象在许多行为中选择一种行为

C. 模板方法在接口的一个方法中定义算法的骨架，将一些步骤的实现延迟到子类中，使得子类可以在不改变算法结构的情况下，重新定义算法中的某些步骤

D. 开闭原则的含义是一个软件实体应当对继承开放，对实现关闭

6. 下列说法中错误的是

A. `std::string` 和 `std::vector` 均可以直接通过下标访问元素。

B. 对于 `std::string` 定义的字符串 `str`，可以分别通过 `str.begin()` 和 `str.end()` 获取迭代器，指向 `str` 的第一个和最后一个字符。

C. 当 `std::vector` 的 `size` 达到 `capacity`，仍然可以继续调用 `push_back()` 函数添加元素。

D. `std::vector` 在大小发生改变时，可能致使所有迭代器失效。

7. 阅读代码，下列说法正确的是

```

#include<iostream>
using namespace std;

void f(int& x) {
    x += 1;
}

void f(int&& x) {
    int& y = x;           // (1)
    const int& z = y + 1;  // (2)
    f(y);                 // (3)
}

int main() {
    int a = 1;
    f(a);                 // (4)
    f(1);                 // (5)
    return 0;}

```

A. 这段代码实现了对函数 f 的重载。

B. 左值引用无法绑定右值，但是常量左值引用可以绑定右值。因此 (1) 的写法有误，(2) 的写法正确。

C. (3) (4) 处为右值引用，调用函数 `void f(int& x)`；(5) 处为左值引用，调用函数 `void f(int&& x)`。

D. 执行 (4) 的函数调用不会影响主函数中变量 a 的取值，即执行完 (4) 后，主函数中 a 的值仍然为 1。

8. 下列说法错误的是

A. 如果类中没有定义任何构造函数，也没有显式删除默认构造函数，则编译器会合成默认构造函数

B. 全局对象在main()函数调用之前构造，在main()函数执行完之前不会被析构

C. 使用全局对象可能会使得代码之间的耦合性增大，因此尽量少用全局变量

D. 假设类A的对象a是类B的数据成员，则类B的对象b执行析构时，会先调用类A的析构函数来析构a，再调用类B的析构函数。

提交格式

请你提交一个文本文件，第*i*行是一个由A/B/C/D组成的字符串，代表第*i*题的答案。

若你不想提交第*i*题，请将第*i*行留空。

你可以通过测试点的详细信息，看到评测对你提交文件的解析。若存在问题，请联系助教。

以下给出一个合法的提交答案的例子：

A
B
C
D
A
B
C
D

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	answer	cp		1048576 B

递交历史

#	状态	时间
表中没有数据		

递交答案

选择文件 未选择任何文件

添加

提交