

📋 题目列表 (/course/46/contest/559/home)

📊 提交状态 (/course/46/contest/559/submissions)

🏆 排行榜 (/course/46/contest/559/ranklist/normal)

## 选择题 第三次

### 选择题

刷新 ↻

如无特殊说明，所有题目的编译选项都包含 `-std=c++11`

1. 【单选题】关于常量成员函数和常量对象的描述，以下哪个选项是正确的？
  - A. 常量成员函数可以修改对象的任何数据成员，而常量对象的数据成员在整个生命周期内不可更改
  - B. 常量对象的数据成员在整个生命周期内不可更改，而常量成员函数不能修改对象的任何数据成员
  - C. 常量成员函数不能修改对象的任何数据成员，但可以调用对象的其他非常量成员函数
  - D. 常量成员函数不能修改对象的任何数据成员，也不能调用对象的其他非常量成员函数
2. 【单选题】如果用户定义了自己的拷贝构造函数，下列哪些成员函数/运算符，编译器将不会自动合成（）
  - A. 移动构造函数
  - B. 默认构造函数
  - C. 拷贝赋值运算符
  - D. 析构函数
3. 【单选题】以下关于面向对象编程和类的继承的说法正确的是（）
  - A. 在private继承中，派生类的成员函数无法访问基类的private成员，也不能访问protected成员
  - B. 在private继承中，即使使用using关键字，基类访问权限为protected的成员也不能被由派生类定义的对象访问
  - C. 在public继承中，基类的private成员不会成为派生类的private成员，派生类无法直接访问基类的private成员
  - D. 在继承中，基类的构造函数、析构函数都不能被派生类继承
4. 【多选题】选出正确的选项

```

void f(int& x) {
}

void f(int&& x) {
    int& y = x; //(1)
    const int& z = y+1; //(2)
    int& w=2; //(3)
    f(z); //(4)
}

```

- A. (1)处会发生编译错误，因为左值引用无法绑定右值
- B. (2)处不会发生编译错误，因为常量左值引用可以绑定右值
- C. (3)处会发生编译错误，因为非const引用不能绑定到右值
- D. (4)处会发生编译错误，因为传递了const引用
5. 【多选题】 以下哪些描述是正确的？

```

class Base {
public:
    virtual void print() {
        cout << "Base class" << endl;
    }
};

class Derived : public Base {
public:
    void print() override {
        cout << "Derived class" << endl;
    }
};

int main() {
    Base* base = new Base();
    Derived* derived = new Derived();

    Base* bp = dynamic_cast<Base*>(derived);
    Derived* dp = static_cast<Derived*>(base);

    bp->print();
    dp->print();

    delete base;
    delete derived;

    return 0;
}

```

- A. bp->print() 会输出 "Base class"。
- B. bp->print() 会输出 "Derived class"。
- C. 代码中 dynamic\_cast 的使用是安全的。
- D. 代码中 static\_cast 的使用是安全的。
6. 【多选题】 关于下面代码说法正确的有：

```

1  #include <iostream>
2  using namespace std;
3
4  class Base0{
5  public:
6      Base0(){};
7      Base0(int d){};
8      void f(double d) { cout << "Base0::f(" << d << ")\n"; }
9      void g(double d) { cout << "Base0::h(" << d << ")\n"; }
10 };
11
12 class Base1: public Base0{
13 public:
14     using Base0::f;
15     void f() { cout << "Base1::f()\n"; }
16     void f(int i) { cout << "Base1::f(" << i << ")\n"; }
17 };
18
19 class Derive : public Base1{
20 public:
21     using Base1::f;
22     void f(int i) { cout << "Derive::f(" << i << ")\n"; }
23 };
24
25 int main() {
26     Derive d;
27     d.f(10);
28     d.f(4.9);
29     d.g(4.8);
30     d.f();
31     return 0;
32 }

```

- A. 如果将6行注释掉, 程序将不能通过编译。
  - B. 如果将第14行注释掉, 程序仍能通过编译,但是输出有变化。
  - C. 如果将第21行注释掉, 程序仍能通过编译,但是输出有变化。
  - D. 如果将第12行改为 `class Base1: protected Base0{`, 只需要去掉第29行就可以成功编译运行。
7. 【多选题】 以下哪些描述是正确的?

```

#include <iostream>
using namespace std;

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

class Derived : public Base {
public:
    void func1() override {
        cout << "Derived::func1()" << endl;
    }
    void func2() {
        cout << "Derived::func2()" << endl;
    }
};

int main() {
    Base *basePtr = new Derived();
    basePtr->func1();
    basePtr->func2();
    delete basePtr;
    return 0;
}

```

- A. basePtr->func1(); 调用的是 Derived::func1()
  - B. basePtr->func2(); 调用的是 Derived::func2()
  - C. func1() 在 Derived 类中是对 Base 类的 func1() 的重写
  - D. func2() 在 Derived 类中是对 Base 类的 func2() 的重写
8. 【多选题】选出**错误**的选项：

```
class MyClass {
    int *ptr;

public:
    MyClass(int val) {
        ptr = new int;
        *ptr = val;
    }

    MyClass(const MyClass &obj) {
        ptr = new int;
        *ptr = *obj.ptr;
    }

    ~MyClass() {
        delete ptr;
    }
};
```

- A. 拷贝构造函数是用于创建一个新对象作为现有对象的副本的特殊构造函数。
- B. 拷贝构造函数只在使用赋值操作符时被调用。
- C. 在这段代码中，拷贝构造函数防止了浅拷贝的发生，从而避免了潜在的内存泄漏和数据访问错误。
- D. 在这段代码中，显式定义拷贝构造函数是不必要的，因为编译器提供的默认拷贝构造函数会正确地拷贝所有成员。

#### 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

#### 递交答案

选择文件

未选择任何文件

添加

提交