



# 习题课



Tank



**1** 以下说法正确的是：

- A) 每个对象内部都有成员函数的实现代码
- B) 一个类的私有成员函数内部不能访问本类的私有成员变量
- C) 类的成员函数之间可以互相调用
- D) 编写一个类时，至少要写一个成员函数



## 2 假设有以下类 A:

```
class A {  
    public:  
    int func(int a) { return a * a; }  
};
```

以下程序片段，哪个是不正确的？

- A) `A a; a.func(5);`
- B) `A * p = new A; p->func(5);`
- C) `A a; A & r = a; r.func(5);`
- D) `A a, b; if( a != b) a.func(5);`



**4** 以下说法中正确的是：

- A) 一个类一定会有无参构造函数
- B) 构造函数的返回值类型是 `void`
- C) 一个类只能定义一个析构函数，但可以定义多个构造函数
- D) 一个类只能定义一个构造函数，但可以定义多个析构函数



**5** 对于通过 new 运算符生成的对象

- A) 在程序结束时自动析构
- B) 执行 delete 操作时才能析构
- C) 在包含该 new 语句的函数返回时自动析构
- D) 在执行 delete 操作时会析构，如果没有执行 delete 操作，则在程序结束时自动析构



- 6** 如果某函数的返回值是个对象，则该函数被调用时，返回的对象
- A) 是通过复制构造函数初始化的
  - B) 是通过无参数的构造函数初始化的
  - C) 用哪个构造函数初始化取决于函数中 `return` 语句是怎么写的
  - D) 不需要初始化

## 7 写出运行结果

```
#include <iostream>
using namespace std;
class CSample {
int x;
public:
CSample() { cout << "C1" << endl; }
CSample(int n ) {
x = n;
cout << "C2,x=" << n << endl; }
};
int main(){
CSample array1[2];
CSample array2[2] = {6,8};
CSample array3[2] = {12};
CSample * array4 = new CSample[3];
return 0;
}
```

Tankertanker Design





## 9(1)

如果将运算符 “[]” 重载为某个类的成员运算符（也即成员函数），则该成员函数的参数个数是：

- A) 0 个 B) 1 个 C) 2 个 D) 3 个

## 9(2)

2. 如果将运算符 “\*” 重载为某个类的成员运算符（也即成员函数），则该成员函数的参数个数是：

- A) 0 个 B) 1 个 C) 2 个 D) 0 个 1 个均可



编写一个**CDemo** 类，包含一个保护**int**类型成员，对++和--运算符进行前置和后置的重载（对成员变量进行操作），其中++重载为成员函数，--重载为全局函数。

WPS PDF编辑试用

# 10

下面的 MyInt 类只有一个成员变量。MyInt 类内部的部分代码被隐藏了。假设下面的程序能编译通过，且输出结果是：

4, 1

请写出被隐藏的部分。（您写的内容必须是能全部放进 MyInt 类内部的，MyInt 的成员函数里不允许使用静态变量）。

```
#include <iostream>
```

```
using namespace std;
```

```
class MyInt {  
    int nVal;  
    public:  
        MyInt( int n) { nVal = n ;}  
        int ReturnVal() { return nVal;}  
};
```

```
int main () {  
    MyInt objInt(10);  
    objInt-2-1-3;  
    cout << objInt.ReturnVal();  
    cout <<",";    objInt-2-1;  
    cout << objInt.ReturnVal();  
    return 0;  
}
```

```
/*
```

```
MyInt & operator-(int n) {  
    nVal -= n;  
    return * this;  
}
```

```
*/
```

Tankertanker Design

# 10

下面的 MyInt 类只有一个成员变量。MyInt 类内部的部分代码被隐藏了。假设下面的程序能编译通过，且输出结果是：

4, 1

请写出被隐藏的部分。（您写的内容必须是能全部放进 MyInt 类内部的，MyInt 的成员函数里不允许使用静态变量）。

```
#include <iostream>
```

```
using namespace std;
```

```
class MyInt {
```

```
    int nVal;
```

```
    public:
```

```
        MyInt( int n) { nVal = n ;}
```

```
        int ReturnVal() { return nVal;}
```

```
};
```

```
int main () {
```

```
    MyInt objInt(10);
```

```
    objInt-2-1-3;
```

```
    cout << objInt.ReturnVal();
```

```
    cout <<",";    objInt-2-1;
```

```
    cout << objInt.ReturnVal();
```

```
    return 0;
```

```
}
```

```
/*
```

```
MyInt & operator-(int n) {
```

```
    nVal -= n;
```

```
    return * this;
```

```
}
```

```
*/
```

Tankertanker Design

## 8

8. 下面程序输出的结果是

4, 6

请填空:

```
class A {  
    int val;  
public:  
    A( int n) { val = n; }  
    int GetVal() { return val;}  
};  
class B: public A {  
    private:  
        int val;  
    public:  
        B(int n): _____{ }  
        int GetVal() { return val;}  
};  
int main() {  
    B b1(2);  
  
    cout<<b1.GetVal()<< ", "  
        << b1.A::GetVal()<< endl;  
    return 0;  
}
```

**A(3\*n), val(2\*n)**

Tankertanker Design



**11** 以下说法不正确的是（假设在公有派生情况下）

- A) 可以将基类对象赋值给派生类对象
- B) 可以将派生类对象的地址赋值给基类指针
- C) 可以将派生类对象赋值给基类的引用
- D) 可以将派生类对象赋值给基类对象

## 12 写出下面程序的输出结果:

```
#include <iostream >
using namespace std;
class B {
public:
    B(){ cout << "B_Con" << endl; }
    ~B() { cout << "B_Des" << endl; }
};
class C:public B {
public:
    C(){ cout << "C_Con" << endl; }
    ~C() { cout << "C_Des" << endl; }
};
int main() {
    C * pc = new C;
    delete pc;
    return 0;
}
```

Tankertanker Design