

# 类和对象

6 questions

1. 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

- ☐ 不需要初始化
- ☐ 是通过无参数的构造函数初始化的
- ☐ 用哪个构造函数初始化取决于函数中 `return` 语句是怎么写的
- ☒ 是通过复制构造函数初始化的

2. 以下说法正确的是：

- ☐ 静态成员变量每个对象有各自的一份
- ☐ `const`成员函数不能作用于非 `const` 对象
- ☐ 在静态成员函数中可以调用同类的其他任何成员函数
- ☒ 在静态成员函数中不能使用 `this` 指针

3. 以下关于 `this` 指针的说法中不正确的是：

- ☒ `const`成员函数内部不可以使用`this` 指针
- ☐ 在构造函数内部可以使用`this`指针
- ☐ 成员函数内的`this` 指针，指向成员函数所作用的对象
- ☐ 在析构函数内部可以使用 `this` 指针

4. 以下程序，哪个是不正确的？

☒

```
int main(){
    class A { int v; };
    A a; a.v = 3; return 0;
}
```

☐

```
int main() {
    class A { public: int v; A * p; };
    A a; a.p = &a; return 0;
}
```

☐

```
int main() {
    class A { public: int v; };
    A * p = new A;
    p->v = 4; delete p;
    return 0;
}
```

☐

```
int main() {
    class A { public: int v; A * p; };
    A a; a.p = new A; delete a.p;
    return 0;
}
```

5. 以下说法中正确的是：

☒

一个类只能定义一个析构函数，但可以定义多个构造函数

☐

一个类一定会有无参构造函数

☐

构造函数的返回值类型是 void

☐

一个类只能定义一个构造函数，但可以定义多个析构函数

6. 对于通过 new 运算符生成的对象

☒

执行 delete 操作时才能析构

☐

在程序结束时自动析构

☐

在包含该 new 语句的函数返回时自动析构

☐

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