

## C++面向对象程序设计模拟试题六

一、单项选择题 (本大题共 10 小题, 每小题 2 分, 共 20 分) 在每小题列出的四个备选项中, 只有一个是符合题目要求的。请将其代码填写在题后的括号内。错选, 多选或未选均无分。

1. 类的析构函数( )。  
A) 作为类的一般成员函数                      B) 类初始化时被调用  
C) 对象初始化时被调用                      D) 对象被删除时被调用
2. 一个类的友元函数或友元类可以访问该类的( )。  
A) 私有成员              B) 保护成员              C) 公有成员              D) 所有成员
3. 下列关于成员函数特征的描述中, ( ) 是错误的。  
A) 成员函数一定是内联函数                      B) 成员函数可以重载  
C) 成员函数可以设置参数的默认值              D) 成员函数可以是静态的
4. 下列函数中, ( ) 不是类的成员函数。  
A) 构造函数              B) 析构函数              C) 友元函数              D) 拷贝构造函数
5. 下列对派生类的描述中, ( ) 是错误的。  
A) 一个派生类可以作为另一个派生类的基类  
B) 派生类至少有一个基类  
C) 派生类的成员除了它自己的成员外, 还包含了它的基类的成员  
D) 派生类中继承的基类成员的访问权限到派生类中保持不变
6. 下列的描述中, ( ) 是错误的。  
A) 公有继承时基类中的 public 成员在派生类中仍是 public 的  
B) 公有继承时基类中的 private 成员在派生类中仍是 private 的  
C) 公有继承时基类中的 protected 成员在派生类中仍是 protected 的  
D) 私有继承时基类中的 public 成员在派生类中仍是 private 的
7. 下列虚基类的声明中正确的是( )。  
A) class virtual B: public A                      B) virtual class B: public A  
C) class B: public A virtual                      D) class B: virtual public A
8. 若类 A 和类 B 的定义如下:

```
class A
{
    int i, j;
public:
    A(int m, int n): i(m), j(n) {}
    int Geti() { return i; }
};

class B: public A
{
    int k;
public:
    B(int m, int n, int u): A(m, n), k(u) {}
    void Make() { k = i * j; }
```

```
};
```

```
int main()
{
    B b(1, 2, 3);
    return 0;
}
```

则上述定义中, ( ) 是非法的表达式.

A)  $k=i*j$ ;      B) `int k;`      C) `return i;`      D) `void Make()`

9. 若有以下程序:

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

```
class A
{
public:
    int a;
    A() { a=10; }
};
```

```
class A1: public A
{
public:
    A1() { a = a + 1; } \
};
```

```
class A2: public A
{
public:
    A2() { a = a + 2; }
};
```

```
class B: public A1, public A2
{
public:
    B() {}
    void Print() { cout << a << endl; }
};
```

```
int main()
{
    B obj;
    obj.Print();
    return 0;
}
```

}

则程序编译或运行后的输出结果为:

A) 提示语法错误    B) 13    C) 12    D) 10

10. 在下面的 4 个关键字中, ( ) 是用来说明虚函数的。

A) virtual    B) public    C) protected    D) private

**二、填空题(本大题共 5 小题,每小题 2 分,共 10 分)将正确的答案写在每小空的空格内,错填或不填均无分。**

1. 通过类创建 ( ) 来调用构造函数。

2. 假定 AB 为一个类, 则执行 “AB a[10];” 语句时, 系统自动调用该类的构造函数的次数为 ( ) 。

3. 使用函数模板的方法是先说明函数模板, 然后实例化成相应的 ( ) 进行调用执行。

4. 拷贝构造函数用它所在类的 ( ) 作为参数。

5. 重载运算符 “<<” 的函数名为 ( ) 。

**三、程序分析题 (本大题共 6 小题, 每小题 5 分, 共 30 分) 给出下面各程序的输出结果。**

1. 若有以下程序:

```
#include <iostream>
```

```
using namespace std;
```

```
class A
```

```
{
```

```
    int a;
```

```
public:
```

```
    A(int aa = 0 )
```

```
    {
```

```
        a = aa;
```

```
        cout << "A(): " << a << endl;
```

```
    }
```

```
};
```

```
class B: public A
```

```
{
```

```
    int b;
```

```
public:
```

```
    B(int aa = 0, int bb = 0): A(aa)
```

```
    {
```

```
        b = bb;
```

```
        cout << "B(): " << b << endl;
```

```
    }
```

```
};
```

```
int main()
```

```
{
```

```
        B x(5), y(6,7);

        return 0;
    }

```

输出结果为:

2. 若有以下程序:

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

```

```
class Point
{
private:
    int x, y;

public:
    Point(int m = 0, int n = 0) { x = m; y = n; }
    void Show() const { cout << "x=" << x << ", " << "y=" << y << endl; }
};

```

```
int main()
{
    Point *p = new Point(1, 68);
    p->Show();
    delete p;

    return 0;
}

```

输出结果为:

3. 若有以下程序:

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

```

```
class Sample
{
private:
    int n;

public:
    Sample(int i) { n = i; s += n; }
    static int s;
}

```

```
        void Show() const { cout << s << endl; }  
};
```

```
int Sample::s = 0;
```

```
int main()  
{  
    Sample a(2), b(5), c(8);  
    c.Show();  
  
    return 0;  
}
```

输出结果为:

4. 若有以下程序:

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

```
class Sample  
{  
    int n;  
public:  
    Sample(int i) { n = i; }  
    void Print() { cout << "1:n=" << n << ", "; }  
    void Print() const { cout << "2:n=" << n << endl; }  
};
```

```
int main()  
{  
    Sample a(10);  
    const Sample b(20);  
    a.Print();  
    b.Print();  
  
    return 0;  
}
```

输出结果为:

5. 若有以下程序:

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

```

class Test
{
    int x;
public:
    void Setx(int i) { x = i; }
    int Putx() { return x; }
};

int main()
{
    Test *p;
    Test a[3];
    a[0].Setx(5);
    a[1].Setx(6);
    a[2].Setx(7);
    for( int j = 0; j < 3; j++)
    {
        p = &a[j];
        cout << p->Putx() << ",";
    }
    cout << endl;

    return 0;
}
输出结果为:

```

6. 若有以下程序:

```

#include <iostream>
using namespace std;

class Base
{
public:
    Base(int x) { a = x; }
    void Show() { cout << a << endl; }
private:
    int a;
};

class Derived: public Base
{
public:
    Derived (int i): Base(i + 1), b(i) {}

```

```

        void Show() { cout << b << endl; }
private:
    int b;
};

int main()
{
    Base b(5), *pb;
    Derived d(1);
    pb = &d;
    pb->Show();

    return 0;
}

```

输出结果为:

#### 四、完成程序填空题（本大题共 4 个小题，每小题 3 分，共 12 分）下面程序都留有空白，请将程序补充完整。

1. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

```

```

class A
{
private:
    static int n;

public:
    A(){ n++; }
    static void Show() { cout << "共有" << n << "个对象!" << endl; }
};

```

\_\_\_\_\_ [1] \_\_\_\_\_ n = 0;                      // 为静态数据成员赋初值

```

int main()
{
    A obj1, obj2, obj3;
    A::Show();

    return 0;
}

```

2. 下列程序的输出结果为“2”，试将程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
public:
    _____[2]_____ Show() const { cout << "1" << endl; }
};

class B: public A
{
public:
    void Show() const { cout << "2" << endl; }
};

int main()
{
    A *p = new B;
    p->Show();
    delete p;

    return 0;
}

```

3. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class Base
{
private:
    int m;

public:
    Base(int a): m(a){ }
    virtual void Show() const { cout << m << endl; }
};

class Derived: public Base
{
private:
    int n;

public:

```



```

Derived(int a, int b): Base(a), n(a) { }
void Show() const
{
    cout << n << ", ";
    _____[3]_____ Show();    // 调用基类的 Show()
}
};

```

```

int main()
{
    Derived obj(158, 98);
    Base *p = &obj;
    p->Show();

    return 0;
}

```

4. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

```

```

class Int
{
private:
    int i;

public:
    Int(int x = 0): i(x) { }
    _____() { return i; } // 类类型转换函数, 将类 Int 转换为基本类型 int
};

```

```

int main()
{
    Int a;
    cout << a << endl;

    return 0;
}

```

## 五、编程题 (本大题共 2 小题, 第 1 小题 12 分, 第 2 小题 16 分, 共 28 分)

1. 设计一个类 Rect, 要求如下:

- (1) 该类中的私有数据成员 length, width 存放它的长和宽, 并且设置它们的默认值是 0。
- (2) 通过成员函数设置其长和宽, 并确保长和宽都在(0, 50)范围之内。

(3) 实现求周长函数 `GetPerimeter()`。

2. 定义一个二维坐标类 `Vector2d`, 二个数据成员为 `double` 型 `x, y` 为 `private` 属性。定义二个参数的构造函数和一个 `Show()` 函数用以输出 `x, y` 的值, 另外作为成员函数重载的运算符“`+`”的功能是将此类二个对象的数据成员 `x` 和 `y` 对应相加。这些成员函数的属性均为 `public`。请用 C++ 编写此程序, 并编写测试程序进行测试

## C++面向对象程序设计模拟试题六参考答案

一、单项选择题 (本大题共 10 小题, 每小题 2 分, 共 20 分) 在每小题列出的四个备选项中, 只有一个是符合题目要求的。请将其代码填写在题后的括号内。错选, 多选或未选均无分。

- |       |       |       |       |         |
|-------|-------|-------|-------|---------|
| 1. D) | 2. D) | 3. A) | 4. C) | 5. D)   |
| 6. B) | 7. D) | 8. A) | 9. A) | 10. A). |

二、填空题 (本大题共 5 小题, 每小题 2 分, 共 10 分) 不写解题过程, 将正确的答案写在每小题的空格内, 错填或不填均无分

1. 参考答案: 对象
2. 参考答案: 10
3. 参考答案: 模板函数
4. 参考答案: 引用
5. 参考答案: operator <<

三、程序分析题 (本大题共 6 小题, 每小题 5 分, 共 30 分) 给出下面各程序的输出结果。

1. 参考答案:  
A(): 5  
B(): 0  
A(): 6  
B(): 7
2. 参考答案: x=1, y=68
3. 参考答案: 15
4. 参考答案: 1:n=10, 2:n=20
5. 参考答案: 5, 6, 7
6. 参考答案: 2

四、完成程序填空题 (本大题共 4 个小题, 每小题 3 分, 共 12 分) 下面程序都留有空白, 请将程序补充完整。

1. 参考答案: [1] int A::
2. 参考答案: [2] virtual void
3. 参考答案: [3] Base::
4. 参考答案: [4] operator int

五、编程题 (本大题共 2 小题, 第 1 小题 12 分, 第 2 小题 16 分, 共 28 分)

1. 参考程序:

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

class Rect
{
private:
    double length, width;
```

```

public:
    Rect(double l = 0, double w = 0): length(l), width(w){ }
    void Set(double l, double w)
    {
        if (length <= 0 || length >= 50 || width <= 0 || width >= 50 )
            throw "数据不在指定范围(0,50)!";

        length = l;
        width = w;
    }
    double GetPerimeter() { return 2 * (length + width); }
};

```

```

int main()
{
    try
    {
        Rect obj(1, 8);
        cout << "周长:" << obj.GetPerimeter() << endl;
    }
    catch (char *str)
    {
        cout << "异常信息:" << str << endl;
    }

    return 0;
}

```

## 2. 参考程序:

```

#include <iostream>
using namespace std;

```

```

class Vector2d
{
    double x, y;
public:
    Vector2d(double a, double b): x(a), y(b) {}
    void Show() { cout << "(" << x << ", " << y << ")" << endl; }
    Vector2d operator+(Vector2d &obj);
};

```

```

Vector2d Vector2d::operator+(Vector2d &obj)
{ return Vector2d(x + obj.x, y + obj.y); }

```

```
int main()
{
    Vector2d d1(3.5, 4.5), d2(2.5, 5.5), d3(0.0, 0.0);
    d3 = d1 + d2;
    d3.Show();

    return 0;
}
```