

C++面向对象程序设计模拟试题三

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。

- 下面有关重载函数的描述中, 正确的是 ()。
A) 重载函数必须具有不同的返回值类型 B) 重载函数形参个数必须不同
C) 重载函数一般具有不同的形参表 D) 重载函数名可以不同
- 假定 MyClass 为一个类, 那么下列的函数说明中, () 为该类的析构函数。
A) void~MyClass(); B) ~MyClass(int n);
C) MyClass(); D) ~MyClass();
- () 是以一种完全通用的方法来设计函数或类而不必预先说明将被使用的每个对象的类型。
A) 模板 B) 类 C) 对象 D) 函数
- 下面关于静态成员函数的述术中错语的是 ()。
A) 静态成员函数可以有返回值 B) 静态成员函数含有 this 指针
C) 静态成员函数可以具有指针参数 D) 静态成员函数可有具有返回值类型
- 定义类模板时要使用关键字 ()。
A) private B) public C) delete D) template
- 下列是重载乘法运算符的函数原型声明, 其中错误的是 ()。
A) friend MyClass operator*(double,double);
B) friend MyClass operator*(double,MyClass);
C) friend MyClass operator*(MyClass,double);
D) friend MyClass operator*(MyClass,MyClass);
- 以下 () 成员函数表示纯虚函数。
A) virtual int fun(int) B) void fun(int)=0
C) virtual void fun()=0 D) virtual void fun(int){}
- 关于纯虚函数, 下列表述中正确的是 ()。
A) 纯虚函数的声明不一定以 "=0" 结束 B) 含有纯虚函数的类可以定义对象
C) 含有纯虚函数的类是抽象类 D) 上面都是错误的
- () 解决二义性问题。
A) 只能使用作用域分辨操作符 B) 只能使用虚基类
C) 使用作用域分辨操作符或虚基类 D) 上面都不能
- 实现编译时的多态性要使用 ()。
A) 重载函数 B) 析构函数 C) 构造函数 D) 虚函数

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

- 定义一个函数名为 fun, 返回值为 int, 没有参数的纯虚函数的声明是 ()。
- 定义静态成员函数应使用关键字 ()。
- 在 C++ 中, 如果类的名称是 CTest, 则此类的构造函数名称为 ()。
- 在类的对象被创建时, () 函数会被自动调用。

5. 用 class 声明的类中数据成员和成员函数的默认访问级别为 ()。

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. 阅读下面程序，写出输出结果。

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

class A
{
public:
    A() { cout << "A()" << endl; }
    ~A() { cout << "~A()" << endl; }
};

int main()
{
    A a, *p = new A;
    delete p;
    cout << "end" << endl;

    return 0;
}
```

上面程序的输出结果为：

2. 阅读下面程序，写出输出结果。

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

class Array
{
private:
    int *elem;
    int size;

public:
    Array(int a[], int sz): elem(a), size(sz){ }
    int GetSize() const { return size; }
    int &operator[](int pos){ return elem[pos - 1]; }
};

int main()
{
    int a[] = {1, 2, 3, 4, 5};
    Array ar(a, 5);
}
```

```

        ar[1] = 8;
        for (int i = 1; i <= ar.GetSize(); i++)
            cout << ar[i] << " ";
        cout << endl;

        return 0;
}

```

上面程序的输出结果为:

3. 阅读下面程序, 写出输出结果。

```

#include <iostream>
using namespace std;

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

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

void Fun1(A obj) { obj.Show(); }
void Fun2(const A &obj) { obj.Show(); }

int main()
{
    B b;
    Fun1(b);
    Fun2(b);

    A *p = new B;
    p->A::Show();
    p->Show();

    return 0;
}

```

上面程序的输出结果为:

4. 阅读下面程序，写出输出结果。

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

template <class ElemType>
ElemType f(ElemType x)
{
    return x * (ElemType)5.1;
}

template <class ElemType>
void f(ElemType x, ElemType y)
{
    x = x * 1.1;
    y = y * 1.1;
    cout << x << endl;
    cout << y << endl;
}

int main()
{
    cout << f(3) << endl;
    f(1, 2);
    f(1.0, 2.0);

    return 0;
}
```

上面程序的输出结果为：

5. 阅读下面程序，写出输出结果。

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

class A
{
private:
    int a;

public:
    A(int m = 1): a(m) {}
    void Show() { cout << a << endl; }
    void Show() const { cout << a + 5 << endl; }
```

```
};
```

```
int main()
{
    A a, b(8);
    a.Show();
    b.Show();
    const A c, d(8);
    c.Show();
    d.Show();

    return 0;
}
```

上面程序的输出结果为:

6. 阅读下面程序, 写出输出结果。

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

class Test
{
private:
    static int n;

public:
    Test() { n += 6; }
    ~Test() { n -= 8; }
    static int GetNum() { return n; }
};

int Test::n = 2;

int main()
{
    cout << Test::GetNum() << endl;
    Test *p = new Test;
    cout << Test::GetNum() << endl;
    delete p;
    cout << Test::GetNum() << endl;
    cout << "end" << endl;

    return 0;
}
```

上面程序的输出结果为:

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. 请完成下面的程序

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

class Test
{
private:
    ____[1]____ int count;           // 对象数

public:
    Test(){ count++; }
    static void Show(){ cout << "共有" << count << "个对象!" << endl; }数
};

int Test::count = 0;

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

    return 0;
}
```

2. 请完成下面的程序

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

class Integer
{
private:
    int a;

public:
    Integer(int x = 0){ a = x; }
    operator int() { return a; }
    Integer operator+(const Integer &i) const
    {
        return ____[2]____;           // 返回和
    }
}
```

```

    }
};

int main()
{
    Integer a(2), b(3), c;
    c = a + b;
    cout << c << endl;

    return 0;
}

```

3. 请完成下面的程序

```

#include <iostream>
using namespace std;

template <class ElemType>
class A
{
private:
    ElemType a;

public:
    A(ElemType x): a(x) { }
    void Show() const { cout << a << endl; }
};

int main()
{
    A<__[3]__> obj("Hello, world!");
    obj.Show();

    return 0;
}

```

4. 下列程序的输出结果为“B”，试将程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
public:
    __[4]__ void Show() const { cout << "A" << endl; }
};

```

```

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

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

    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. 试使用函数模板实现求一个数组各元素的平均值，要求编写测试程序。

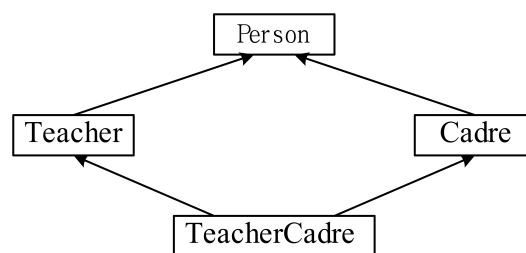
函数模板声明如下：

```
template <class T>
```

```
T Average(T a[], int n);
```

// 求数组 a 各元素之平均值

2. 定义 Person（人）类，由 Person 分别派生出 Teacher（教师）类和 Cadre（干部）类，再由 Teacher（教师）类和 Cadre（干部）类采用多重继承方式派生出新类 TeacherCadre（教师兼干部）类，各类之间的继承关系如下图所示。



要求：

（1）在 Person 类中包含的数据成员有姓名(name)、年龄(age)、性别(sex)。在 Teacher 类还包含数据成员职称(title)，在 Cadre 类中还包含数据成员职务(post)，在 TeacherCadre 类中还包含数据成员工资(wages)。

（2）在类体中定义成员函数。

（3）每个类都有构造函数与显示信息函数(Show)。

C++面向对象程序设计模拟试题三参考答案安全

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

二、填空题 (本大题共 5 小题, 每小题 2 分, 共 10 分) 不写解答过程, 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: `virtual int fun() = 0;`
2. 参考答案: `static`
3. 参考答案: `CTest`
4. 参考答案: 构造函数
5. 参考答案: `private` 或私有

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()
A()
~A()
end
~A()
```

2. 参考答案:

```
8 2 3 4 5
```

3. 参考答案:

```
A
B
A
B
```

4. 参考答案:

```
15
1
2
1.1
2.2
```

5. 参考答案:

```
1
8
6
13
```

6. 参考答案:

```
2
8
```

0

end

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] static
2. 参考答案: [2] a + i.a 或 this->a + i.a
3. 参考答案: [3] char *
4. 参考答案: [4] virtual

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;
```

```
template <class T>
T Average(T a[], int n)
{
    T s = 0;
    for (int i = 0; i < n; i++)
        s = s + a[i];
    return s / n;
}
```

```
int main()
{
    double a[] = {11.8, 2, 3, 4, 5.5, 6.8, 7, 8, 9};
    cout << Average(a, 9) << endl;

    return 0;
}
```

2. 参考程序:

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

```
class Person
{
protected:
    char name[18];
    int age;
    char sex[3];
```

```
public:
```

```

    Person(char nm[], int ag, char sx[])
    {
        strcpy(name, nm);
        age = ag;
        strcpy(sex, sx);
    }
    void Show() const
    {
        cout << "姓名:" << name << endl;
        cout << "年龄:" << age << endl;
        cout << "性别:" << sex << endl;
    }
};

class Teacher: virtual public Person
{
protected:
    char title[18];
public:
    Teacher(char nm[], int ag, char sx[], char tl[]): Person(nm, ag, sx)
    { strcpy(title, tl); }
    void Show() const
    {
        Person::Show();
        cout << "职称:" << title << endl;
        cout << endl;
    }
};

class Cadre: virtual public Person
{
protected:
    char post[18];
public:
    Cadre(char nm[], int ag, char sx[], char pt[]): Person(nm, ag, sx)
    { strcpy(post, pt); }
    void Show() const
    {
        Person::Show();
        cout << "职务:" << post << endl;
        cout << endl;
    }
};

```

```

class TeacherCadre: public Teacher, public Cadre
{
protected:
    double wages;
public:
    TeacherCadre(char nm[], int ag, char sx[], char tl[], char pt[], double wg)
        : Person(nm, ag, sx), Teacher(nm, ag, sx, tl), Cadre(nm, ag, sx, pt)
    { wages = wg; }
    void Show() const
    {
        Person::Show();
        cout << "职称:" << title << endl;
        cout << "职务:" << post << endl;
        cout << "工资:" << wages << "元" << endl;
        cout << endl;
    }
};

int main()
{
    Teacher objTeacher("文冠杰", 48, "男", "教授");
    Cadre objCadre("周杰", 56, "男", "院长");
    TeacherCadre objTeacherCadre("李靖", 50, "女", "教授", "院长", 6890);

    objTeacher.Show();
    objCadre.Show();
    objTeacherCadre.Show();

    return 0;
}

```