C++面向对象程序设计模拟试题四

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. 下列关于 C++函数的说明中，正确的是（　　）。
   A）内联函数就是定义在另一个函数体内部的函数
   B）函数体的最后一条语句必须是 return 语句
   C）调用一个函数之前，如果还没有定义这个函数，必须先声明其原型
   D）编译器会根据函数的返回值类型和参数表来区分函数的不同重载形式
2. 假定 MyCIass 为一个类，那么下列的函数说明中，（　　）为该类的无参构造函数。
   A）void MyClass(); B）~MyClass(int n);
   C）MyClass( ); D）~MyClass( );
3. 下列叙述中，错误的是（　　）。
   A）派生类可以使用 private 派生
   B）对基类成员的访问必须是无二义性的
   C）基类成员的访问能力在派生类中维持不变
   D）赋值兼容规则也适用于多继承的组合
4. 当一个类的某个函数被说明为 virtual 时，与这个函数的原型相同的函数在该类及其所有派生类中（　　）。
   A）都是虚函数 B）只有被重新说明为 vittual 时才是虚函数
   C）都不是虚函数 D）上面都不正确
5. 派生类的构造函数的成员初始化表中，不能包含（　　）。
   A）基类的构造函数 B）派生类中子对象的初始化
   C）基类中子对象的初始化 D）派生类中一般数据成员的初始化
6. 下列是重载加法运算符的函数原型声明，其中错误的是（　　）。
   A）MyClass operator+(double,double);
   B）MyClass operator+(double,MyClass);
   C）MyClass operator+(MyClass,double);
   D）MyClass operator+(MyClass,MyClass);
7. 派生类中的成员不能直接访问基类中的（　　）成员。
   A）public B）private C）virtual D）protected
8. 实现运行时的多态性要使用（　　）。
   A）重载函数 B）析构函数 C）构造函数 D）虚函数
9. 如果在类 MyClass 外面的非成员函数中使用了函数调用 MyClass::f()；则函数 f()是类 MyClass 的（　　）。
   A）静态成员函数 B）非静态成员函数
   C）友元函数 D）前面都不正确
10. 由于常对象不能被更新，因此 C++标准中规定（　　）。
   A）通过常对象可以调用它的常成员函数
   B）通过常对象只能调用静态成员函数
   C）常对象的成员都是常成员
   D）通过常对象可以调用任何不改变对象值的成员函数

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题的空格内。错填或不填均无分。

1. 对于派生类的构造函数，在定义对象时构造函数的执行顺序为：先执行调用_____的构造函数，再执行调用子对象类的构造函数，最后执行派生类的构造函数体中的内容。
2. 声明类模板应使用关键字（　　）。
3. 重载下标运算符“[]”的函数名为（　　）。
3. 重载运算符使用的关键字是（　　）。
4. 在面向对象方法中，类的实例称为（　　）。
5. 在类的对象被释放时，（　　）函数会被自动调用。

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

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

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

int main()
{
    A a; B b;
    Show(a); Show(b);

    A *p;
    p = &a;   p->Show();
    p = &b;   p->Show();

    B *q;
    q = &b;   q->Show();

    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;

template <class ElemType>
void Show(ElemType a[], int n)
{
	for (int i = 0; i < n; i++)
	{ cout << a[i] << "    "; }
}

int main()
{
	int a[] = {1, 6, 9};
	Show(a, sizeof(a) / sizeof(int));
	Show(a, 2);
	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 MyClass
{
public:
	MyClass(){ count++; }
	~MyClass(){   count--; }
	static int GetCount(){ return count; }

private:
	static int count;
};

int MyClass::count = 0;
```

```
int main()
{
	MyClass obj1;
	cout << MyClass::GetCount() << endl;
	MyClass obj2;
	cout << MyClass::GetCount() << endl;
	MyClass obj3;
	cout << obj1.GetCount() << endl;
	MyClass *p = new MyClass;
	cout << MyClass::GetCount() << endl;
	delete p;
	cout << MyClass::GetCount() << endl;

	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;

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

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

int main()
{
	B obj;
	A *p = &obj;
	p->f();
```

```
    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 Sample
{
    int i;
public:
    Sample();
    void Display( );
    ~Sample();
};

Sample::Sample()
{
    cout << "constructor" << ",";
    i=0;
}

void Sample::Display() { cout << "i=" << i << ","; }

Sample::~Sample() { cout << "destructor" << endl; }

int main()
{
    Sample a;
    a.Display();

    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 A
{
```

```
    int a, b;
public:
    A(): a(0), b(0) { cout << a << ' ' << b << endl; }
    A(int aa, int bb): a(aa), b(bb) { cout << a << ' ' << b << endl; }
};

int main()
{
    A x, y(2,3);

    return 0;
}
```

上面程序的输出结果为：

四、完成程序填题（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:
    int a;

public:
    Test(int x = 0):___[1]___{}  // 将 a 初始化为 x
    void Show() const { cout << "a:" << a << endl; }
};

int main()
{
    Test obj(168);
    obj.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; }
    ___[2]___() { return a; }                    // 类型转换函数(将类类型转换为整型)
};

int main()
{
    Integer a = 6;
    cout << a << endl;

    return 0;
}
```

3. 请完成下面的程序

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

class Complex
{
private:
    double realPart;
    double imagePart;

public:
    Complex(double real = 0, double image = 0): realPart(real), imagePart(image){ }
    double GetRealPart() const{ return realPart; }
    double GetImagePart() const{ return imagePart; }
    Complex operator+(const Complex &a) const
    {
        return Complex(___[3]___);                // 返回和
    }
};

int main()
{
    Complex a(1, 2), b(2, 6), c;

    c = a + b;
    cout << "a=" << a.GetRealPart() << "+" << a.GetImagePart() << "i" << endl;
    cout << "b=" << b.GetRealPart() << "+" << b.GetImagePart() << "i" << endl;
```

```
    cout << "c=" << c.GetRealPart() << "+" << c.GetImagePart() << "i" << endl;

    return 0;
}
```

4. 将如下程序补充完整。

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

class Test
{
private:
    static int count;

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

_____[4]_____0;                          // 为静态数据成员赋初值

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

    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最小值，要求编写测试程序。

函数模板声明如下：

```
template <class Type>
Type Min(Type a[], int n);                    // 求数组 a 各元素的最小值
```

2. 编写程序，定义抽象基类 Shape(形状)，由它派生出 3 个派生类: Circle(圆形)、Rectangle(矩形)和 Square (正方形)，用函数函数 ShowArea()分别显示各种图形的面积，最后还要显示所有图形的总面积。

C++面向对象程序设计模拟试题四参考答巡查

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．C）	3．C）	4．A）	5．C）
6．A）	7．B）	8．D）	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：基类
2．参考答案：template
3．参考答案：operator[]
4．参考答案：对象
5．参考答案：析构函数

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：
Class A
Class B
Class A
Class B
Class B
2．参考答案：
1　6　9　1　6
3．参考答案：
1
2
3
4
3
4．参考答案：
A()
B()
B::f()
~B()
~A()
5．参考答案：
constructor,i=0,destructor
6．参考答案：
0 0
2 3

四、完成程序填题（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]a(x)

2．参考答案：[2]operator int

3．参考答案：[3] realPart + a.realPart, imagePart + a.imagePart 或 this->realPart + a.realPart, this->imagePart + a.imagePart

4．参考答案：int Test::count =

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 Type>
Type Min(Type a[], int n)
{
	Type m = a[0];
	for (int i = 1; i < n; i++)
		if (a[i] < m) m = a[i];
	return m;
}

int main()
{
	int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
	cout << Min(a, 9) << endl;

	return 0;
}
```

2．参考程序：

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

const double PI = 3.1415926;

class Shape
{
public:
	Shape() { }
	virtual ~Shape() { }
	virtual void ShowArea() const = 0;
	static double totalArea;
	static void ShowTotalArea() { cout << "总面积:" << totalArea << endl; }
```

```
};

class Circle: public Shape
{
private:
    double radius;

public:
    Circle(double r): radius(r) { totalArea += PI * r * r; }
    ~Circle() { }
    virtual void ShowArea()    const { cout << "圆面积:" << PI * radius * radius << endl; };
};

class Rectangle: public Shape
{
private:
    double length;
    double width;

public:
    Rectangle(double l, double w): length(l), width(w){ totalArea += l * w; }
    ~Rectangle() { }
    virtual void ShowArea() const { cout << "矩形面积:" << length * width << endl; };
};

class Square: public Shape
{
private:
    double length;

public:
    Square(double l): length(l){ totalArea += l * l; }
    ~Square() { }
    virtual void ShowArea() const { cout << "正方形面积:" << length * length << endl; };
};

double Shape::totalArea = 0;

int main()
{
    Circle c(1);
    c.ShowArea();

    Rectangle r(1, 2);
```

```
    r.ShowArea();

    Square z(3);
    z.ShowArea();

    Shape::ShowTotalArea();

    return 0;
}
```