C++面向对象程序设计模拟试题一

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. 说明虚函数的关键字是（　　）。
   A）inline　　B）virtual　　C）define　　D）static
2. 在标准 C++中，每个程序中都必须包含有这样一个函数，该函数的函数名为（　　）。
   A）main　　B）MAIN　　C）name　　D）function
3. cout 是某个类的标准对象的引用，该类是（　　）。
   A）ostream　　B）istream　　C）stdout　　D）stdin
4. 如果在类外的非类的成员函数中有函数调用 CPoint::func()；则函数 func()是类 CPoint 的（　　）。
   A）私有静态成员函数　　B）公有非静态成员函数
   C）公有静态成员函数　　B）友元函数
5. 如果 class 类中的所有成员在定义时都没有使用关键字 public、private 或 protected，则所有成员缺省定义为（　　）。
   A）public　　B）protected　　C）private　　D）static
6. 一个类的所有对象共享的是（　　）。
   A）私有数据成员　　B）公有数据成员
   C）保护数据成员　　D）静态数据成员
7. 动态联编所支持的多态性称为（　　）。
   A）虚函数　　B）继承
   C）编译时多态性　　D）运行时多态性
8. 定义类模板时要使用关键字（　　）。
   A）const　　B）new　　C）delete　　D）template
9. 对虚基类的定义（　　）。
   A）不需要使用虚函数　　B）必须使用虚函数
   C）必须使用 private　　D）必须使用 public
10. 类类型转换函数（　　）。
   A）不能带有参数　　B）只能带一个参数
   C）只能带 2 个参数　　D）只能带 3 个参数

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. 在用 C++进行程序设计时，最好用（　　）代替 malloc。
2. 函数模板中紧随 template 之后尖括号内的类型参数都要寇以保留字（　　）。
3. 编译时多态性可以用（　　）函数实现。
4. 拷贝构造函数用它所在类的（　　）作为参数。
5. 用关键字 static 修饰的类的成员称为(　　)成员。

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 Array
{
public:
	Array(int a[], int iSize):elem(a), size(iSize)
	{
	}

	int GetSize()
	{
		return size;
	}

	int &operator[](int i)
	{
		return elem[i - 1];
	}

private:
	int *elem;
	int size;
};

int main()
{
	int s[]={3, 7, 2, 1, 5};
	Array ar(s, 5);
	ar[1] = 9;
	for (int i = 1; i <= 5; i++)
	{
		cout << ar[i] << "    ";
	}
	cout << 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;
```

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

int main()
{
	int a[] = { 5, 6, 8};
	double b[] = {6.8, 9.6};

	Print(a, sizeof(a) / sizeof(int));
	Print(b, 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 Test
{
public:
	Test(int n):num(n)
	{
		count++;
	}

	~Test()
	{
	}

	void Print() const;

	static int GetCount()
	{
```

```
        return count;
    }

private:
    int num;
    static int count;
};

int Test::count = 0;

void Test::Print() const
{
    cout << this->num << "    " << this->count << "    ";
}

int main()
{
    Test oTest1(6);
    oTest1.Print();

    Test oTest2(8);
    oTest2.Print();

    cout << Test::GetCount();
    cout << 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 Test
{
public:
    Test(int a = 0, int b = 0, int c = 0):x(a), y(b), z(c) {}

    void Print()
    {
        cout << x << endl;
        cout << y << endl;
```

```
    }

    void Print() const
    {
        cout << z << endl;
    }


private:
    int x, y;
    const int z;
};

int main()
{
    Test obj1;
    obj1.Print();

    Test obj2(1, 6, 8);
    obj2.Print();

    const Test obj3(6, 0, 18);
    obj3.Print();

    cout << endl;

    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 MyClass
{
private:
    static int n;

public:
    MyClass() { n += 1; }
    ~MyClass() { n -= 1; }
    static int GetNum() { return n; }
```

```
};

int MyClass::n = 0;

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

    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
{
private:
    int a;

public:
    A() { cout << "无参构造函数" << endl; }
    A(int a) { cout << "含参构造函数 a=" << a << endl; }
    A(const A &copy): a(copy.a) { cout << "复制构造函数" << endl; }
    ~A() { cout << "析构函数" << endl; }
};

int main()
{
    A obj1, obj2(1), obj3(obj2);

    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．将如下程序补充完整。

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

class Test
{
private:
    int num;

public:
    Test(int num = 0) { _____[1]_____ = num; }    //初始化数据成员 num 为形参 num
    int GetNum() const { return num; }
};

int main()
{

    Test obj;
    cout << obj.GetNum() << endl;

    return 0;
}
```

2．将如下程序补充完整。

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

class A
{
private:
    int a;

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

class B: A
{
private:
    int b;
```

```
public:
    B(int m, int n = 0): _____[2]_____ {} // 初始化数据成员 b 的值为 n
    void Show() const
    {
        A::Show();
        cout << b << endl;
    }
};

int main()
{
    B obj(8);
    obj.Show();

    return 0;
}
```

3. 下列程序的输出结果为：

```
0
1
0
```

试将程序补充完整。

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

class Point
{
private:
    int x, y;
    static int count;

public:
    Point(int m = 0, int n = 0): x(m), y(n) { count++; }
    ~Point() { count--; }
    int GetX() const { return x; }
    int GetY() const { return y; }
    static void ShowCount() { cout << count << endl; }
};

_____[3]_____                                    // 静态数据成员的初始化为 0

int main()
{
```

```
    Point::ShowCount();
    Point *p = new Point;
    Point::ShowCount();
    delete p;
    Point::ShowCount();

    return 0;
}
```

4. 将如下程序补充完整。

```
#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; }
    _____[4]_____(const Complex &a) const        // 重载加法运算符+
    {
        Complex b;
        b.realPart = this->realPart + a.realPart;
        b.imagePart = this->imagePart + a.imagePart;

        return b;
    }
};

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;
}
```

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．编写一个函数模板，用于求参数的绝对值，并编写测试程序进行测试。
函数模板声明如下：
template <class Type>
Type Abs(Type a);

2．定义一个复数类 Complex，定义带有 2 个参数（其中一个为缺省参数）的构造函数，显示复数值的函数 Show()， 重载“+”运算符（用成员函数实现），并编写测试程序进行测试。

C++面向对象程序设计模拟试题一参考答案

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．B)　2．A)　3．A)　4．C)　5．C)
6．D)　7．D)　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：内联函数或内置函数
2．参考答案：class 或 typename
3．参考答案：重载
4．参考答案：对象
5．参考答案：静态

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：9 7 2 1 5
2．参考答案：5　6　8　6.8　9.6
3．参考答案：6　1　8　2　2
4．参考答案：
0
0
1
6
18
5．参考答案：
0
1
2
1
end
6．参考答案：
无参构造函数
含参构造函数 a=1
复制构造函数
析构函数
析构函数
析构函数

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] this->num 或 Integer::num
2．参考答案：[2] A(m), b(n)

3．参考答案：[3] int Point::count = 0;
4．参考答案：[4] Complex operator+

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 Abs(Type a)
{
	if (a >= 0) return a;
	else return - a;
}

int main()
{
	cout << Abs(5) << endl;
	cout << Abs(-5) << endl;
	cout << Abs(2.5) << endl;
	cout << Abs(-2.5) << endl;

	return 0;
}
```

2．参考程序：

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

class Complex
{
public:
	Complex(double r, double i = 0)
	{
		real = r;
		image = i;
	}

	void Show()
	{
		cout << real;
		if (image > 0) cout << "+" << image << "i" << endl;
		else if(image < 0) cout << "-" << -image << "i" << endl;
		else cout << endl;
	}
```

```
    Complex operator+(const Complex &obj)
    {
        Complex temp(real + obj.real, image + obj.image);
        return temp;
    }

private:
    double real, image;
};

int main()
{

    Complex z1(2, 6), z2(3, 8), z3(0);

    z1.Show();
    z2.Show();
    z3.Show();

    z3 = z1 + z2;
    z3.Show();

    return 0;
}
```