

## 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。

- 说明虚函数的关键字是（ ）。  
A) inline      B) virtual      C) define      D) static
- 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
- cout 是某个类的标准对象的引用，该类是（ ）。  
A) ostream      B) istream      C) stdout      D) stdin
- 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
- 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
- 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
- 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
- 定义类模板时要使用关键字（ ）。  
A) const      B) new      C) delete      D) template
- 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
- 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。

- 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冠以保留字 ( )。
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 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：new
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] 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;
}

```

## C++面向对象程序设计模拟试题二

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. 说明内联函数的关键字是（ ）。  
A) inline      B) virtual      C) define      D) static
2. 假定 CAb 为一个类，则执行 CAb oX; 语句时将自动调用该类的（ ）。  
A) 有参构造函数      B) 无参构造函数  
C) 拷贝构造函数      D) 赋值重载函数
3. cin 是某个类的标准对象的引用，该类是（ ）。  
A) ostream      B) istream      C) stdout      D) stdin
4. 下面的哪个保留字不能作为函数的返回类型？（ ）。  
A) void      B) int      C) template      D) long
5. 派生类的成员函数不能访问基类的（ ）。  
A) 保护成员      B) 公有成员  
C) 私有成员      D) 前面各选项都正确
6. 在语句 “cout << data;” 中，cout 是（ ）。  
A) C++ 的关键字      B) 类名  
C) 对象名      D) 函数名
7. 编译时多态性使用什么获得？（ ）。  
A) 重载函数      B) 继承      C) 虚函数      D) B 和 C
8. 拷贝构造函数的参数通常是（ ）。  
A) 无特殊要求      B) 指向对象的指针  
C) 本类对象的常引用      D) 对象
9. C++ 有几种联编？（ ）。  
A) 1 种      B) 2 种      C) 3 种      D) 4 种
10. 基类和派生类可以分别称为（ ）。  
A) “大类” 和 “小类”      B) “父类” 和 “子类”  
C) “小类” 和 “大类”      D) “子类” 和 “父类”

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. 设函数 max 是由函数模板实现的，并且 max(3.5, 5) 和 max(3, 5) 都是正确的函数调用，则此函数模板具有（ ）个类型参数。
2. 在 C++ 中，函数重载与虚函数帮助实现了类的（ ）性。
3. 由 static 修饰的数据成员为该类的所有对象（ ）。
4. 重载函数一般在参数类型或参数个数上不同，但（ ）相同。
5. 使用 new 建立的动态对象在不用时应该用（ ）释放所占用的空间。

**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 Point
{
public:
    Point(int a = 0, int b = 0):x(a), y(b) {}
    int GetX() const { return x; }
    int GetY() const { return y; }
    void SetX(int a) { x = a; }
    void SetY(int a) { y = a; }

private:
    int x;
    int y;
};

int main()
{
    Point u;
    const Point v(6, 8);

    cout << u.GetX() << endl;
    u.SetX(16);
    cout << u.GetX() << endl;
    u.SetY(18);
    cout << u.GetY() << endl;

    cout << v.GetX() << endl;
    cout << v.GetY() << 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>
class Test
{
public:
    Test(Type a[], int iSize):elem(a) { size = iSize; }
    void Print() const { for (int i = 0; i < size; i++) cout << elem[i] << " "; }

private:
    Type *elem;
    int size;
};

int main()
{
    int a[] = { 1, 0, 8 };
    double b[] = { 1.6, 1.8 };

    Test<int> ar1(a, 3);
    ar1.Print();

    Test<double> ar2(b, sizeof(b) / sizeof(double));
    ar2.Print();

    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 Goods
{
public:
    Goods(int w): weight(w){ totalWeight = totalWeight + w; }
    Goods(const Goods &g)
    {
        weight = g.weight;
        totalWeight = totalWeight + weight;
    }
    ~Goods() { totalWeight = totalWeight - weight; }
}

```



```

        void Print() const;
        static int GetTotalWeight() { return totalWeight; }

private:
        int weight;
        static int totalWeight;
};

int Goods::totalWeight = 0;

void Goods::Print() const
{
    cout << this->weight << " " << this->totalWeight << " ";
}

int main()
{
    Goods g1(6);
    g1.Print();

    Goods g2(g1);
    g2.Print();

    cout << Goods::GetTotalWeight();
    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;

template <class Type>
class Test
{
public:
    Test(Type a = 0, Type b = 0, Type c = 0):z(c) { x = a; y = b; }
    void Print()
    {
        cout << x << endl;
        cout << y << endl;
    }
}

```

```

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

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

int main()
{
    Test<float> t1;
    t1.Print();

    Test<int> t2(1, 9, 6);
    t2.Print();

    const Test<double> t3(0, 6, 1.8);
    t3.Print();

    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;

template <class Type>
Type Max(const Type &a, const Type &b)
{
    if (a < b) return b;
    else return a;
}

template <class Type>
Type Min(const Type &a, const Type &b)
{
    if (a < b) return a;
    else return b;
}

```

```

int main()
{
    double x = 5.38, y = 6.09;
    cout << Max(x, y) << " " << Max<int>(x, y) << " ";
    cout << Min(x, y) << " " << Min<int>(x, y) << 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
{
public:
    A() { cout << "A" << endl; }
    ~A() { cout << "~A" << endl; }
};

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

int main(void)
{
    B obj;

    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. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
private:
    int n;

public:
    A(int n) { _____[1]_____ = n; } // 将数据成员 n 初始化为形参 n
    void Show() const { cout << n << endl; }
};

```

```

int main()
{

    A i = 8;
    i.Show();

    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
{
public:
    B(int m): _____[2]_____ {} // 将数据成员 a 初始化为 m
    void Show() const { A::Show(); }
};

int main()

```

```

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

    return 0;
}

```

3. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class Test
{
private:
    static int num;

public:
    Test() { num++; }
    ~Test() { num--; }
    static void ShowObjectNum() { cout << num << endl; }
};

```

---

[3] // 静态数据成员的初始化为 0

```

int main(void)
{
    Test::ShowObjectNum();
    Test obj;
    Test::ShowObjectNum();

    return 0;
}

```

4. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class Integer
{
private:
    int i;

public:
    Integer(int x = 0): i(x) { }
}

```

```
_____ [4] _____ { return i; }// 类类型转换函数,将类 Integer 转换为基本类型 int  
};
```

```
int main()  
{  
    Integer a, b(18);  
    cout << a << endl;  
    cout << int(b) << 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，用于求数组中各元素之和，并编写测试程序进行测试。

函数模板声明如下：

```
template <class Type>  
Type Sum(Type a[], int n);
```

2. 定义一个抽象类 Shape，它有一个纯虚函数 GetPerimeter(); 派生出四边型类 Rectangle 和圆类 Circle，在派生类中重载函数 GetPerimeter()，用于求图形的周长，编写测试程序进行测试。

## 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途中错语的是（ ）。  
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) 纯虚函数的声明总是以 "=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（ ）。

3. 在 C++ 中，如果类的名称是 CTest，则此类的构造函数名称为（ ）。
4. 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。

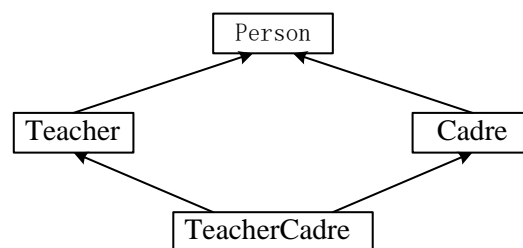
函数模板声明如下：

```

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;
}

```

## 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 `MyClass` 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 `virtual` 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) 通过常对象只能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題的空格內。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:
    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
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]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;
}

```

## C++面向对象程序设计模拟试题五

**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. 下列对类的构造函数和析构函数描述正确的是（ A ）。
  - A) 构造函数可以重载，析构函数不能重载
  - B) 构造函数不能重载，析构函数可以重载
  - C) 构造函数可以重载，析构函数可以重载
  - D) 构造函数不能重载，析构函数不能重载
2. 在函数定义前加上关键字“inline”，表示该函数被定义为（ B ）。
  - A) 重载函数
  - B) 内联函数
  - C) 成员函数
  - D) 普通函数
3. 下面有关重载函数的说明中，（ C ）是正确的。
  - 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列表
  - D) 重载函数名可以不同
4. 下列有关类与对象的说法中，（ C ）是不正确的。
  - A) 对象是类的一个实例
  - B) 任何一个对象只能属于一个具体的类
  - C) 一个类只能有一个对象
  - D) 类和对象的关系和数据类型与变量的关系类似
5. 已知：Print( )函数是一个类的常成员函数，它无返回值，下列表示中，正确的是（ A ）。
  - A) void Print() const;
  - B) const void Print();
  - C) void const Print();
  - D) void Print(const)
6. 假定 Myclass 为一个类,那么下列的函数说明中( D )为该类的析构函数。
  - A) void ~Myclass( );
  - B) ~Myclass( int n);
  - C) Myclass( );
  - D) ~Myclass( )
7. 下面类的定义中有( C )处错误。

```

class myclass
{
    int i=0;
public:
    void myclass( );
    ~myclass(value);
};

```

A) 1            B) 2            C) 3            D) 4

8. 说明虚函数的关键字是 ( B )。

A) inline            B) virtual            C) define            D) static

9. cout 是某个类的标准对象的引用, 该类是 ( A )。

A) ostream            B) istream            C) stdout            D) stdin

10. 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 ( C )。

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

**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空的空格内, 错填或不填均无分。**

1. 重载运算符 “-” 的函数名为( operator- )。

2. C++中类的用途有两种, 一种是类的实例化, 即生成类的对象, 另一种是通过 ( 类的继承 ), 派生出新的类。

3. 派生类中的成员不能直接访问基类中的 ( private ) 成员。

4. 编译时多态性可以用 ( 重载 ) 函数实现。

5. 使用 new 建立的动态对象在不用时应该用 ( delete ) 删除, 以便释放所占空间。

**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; }
    ~A( ) {cout << "Destructor A! " << a << endl; }
};

```

```

class B: public A
{
    int b;

```

```

public:

```

```

        B(int aa = 0, int bb = 0): A(aa) { b = bb; }
        ~B() { cout << "Destructor B! " << b << endl; }
};

```

```

int main()
{
    B x(5), y(6,7);

    return 0;
}

```

上面程序的输出结果为：

2. 若有以下程序：

```

#include <iostream>
using namespace std;

class Point
{
    int x, y;
public:
    Point() { x = 0; y = 0; }
    void SetPoint(int x1, int y1) { x = x1; y = y1; }
    void DisPoint() { cout << "x=" << x << "," << "y=" << y << endl; }
};

int main()
{
    Point *p = new Point;
    p->SetPoint(5, 12);
    p->DisPoint();
    delete p;

    return 0;
}

```

上面程序的输出结果为：

3. 若有以下程序：

```

#include <iostream>
using namespace std;

class Sample
{

```



```

        int n;

public:
    Sample (int i) { n=i; }
    void Add() { s += n; }
    static int s;
    void Dis() { cout << s << endl; }
};

```

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

```

int main()
{
    Sample a(2), b(5), c(8);
    a.Add( );
    b.Add( );
    c.Dis( );

    return 0;
}

```

上面程序的输出结果为：

4. 若有以下程序：

```

#include <iostream>
using namespace std;

```

```

class Base
{
public:
    void Fun() { cout << "1" << endl;}
};

```

```

class Derived:public Base
{
public:
    void Fun() { cout << "2" << endl; }
};

```

```

int main()
{
    Derived a;
    Base *p;
    p=&a;
    p->Fun();
}

```

```

        a.Fun();

        return 0;
}

```

上面程序的输出结果为：

5. 若有以下程序：

```

#include <iostream>
using namespace std;

template<class T1, class T2>
void Fun(T1 &x, T2 &y)
{
    if (sizeof(T1) > sizeof(T2) ) x = (T1)y;
    else y = (T2)x;
}

int main()
{
    double d;
    int i ;
    d = 99.99;
    i = 88;
    Fun(d,i);
    cout << "d=" << d << "i=" << i << 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 Point
{
public:
    Point (int a = 0, int b = 0): x(a), y(b) { }
    int GetX() const { return x; }
    int GetY() const { return y; }
    void SetX(int a) { x = a; }
    void SetY(int a) { y = a; }

private:

```

```

        int x;
        int y;
    };

int main()
{
    Point p1;
    const Point p2(3, 4);
    cout << p1.GetX() << endl;
    p1.SetX(1);
    cout << p1.GetX() << endl;
    p1.SetY(2);
    cout << p1.GetY() << endl;
    cout << p2.GetX() << endl;
    cout << p2.GetY() << 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. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class Test
{
private:
    int n;

public:
    Test(int n) { ____[1]____; }           // 实始化数据成员 n 为形能 n
    void Show() const { cout << n << endl; }
};

int main()
{

    Test i(108);
    i.Show();

    return 0;
}

```

```
}
```

参考答案: this->n 或 Test::n

2. 将如下程序补充完整。

```
#include <iostream>
```

```
using namespace std;
```

```
class A
```

```
{
```

```
private:
```

```
    int a;
```

```
public:
```

```
    A(int x) { a = x; }
```

```
    void Show() const
```

```
    { cout << "a:" << a << endl; }
```

```
};
```

```
class B: public A
```

```
{
```

```
protected:
```

```
    int b;
```

```
public:
```

```
    B(int x, int y):____[2]____{ b = y; }    // 初始化 a 为 x, y 为 b
```

```
    void Show() const
```

```
    {
```

```
        A::Show();
```

```
        cout << "b:" << b << endl;
```

```
    }
```

```
};
```

```
int main()
```

```
{
```

```
    B obj(5, 18);
```

```
    obj.Show();
```

```
    return 0;
```

```
}
```

3. 将如下程序补充完整。

```
#include <iostream>
```

```
using namespace std;
```

```

class Test
{
private:
    static int num;

public:
    Test() { num++; }
    ~Test() { num--; }
    static void ShowObjectNum() { cout << num << endl; }
};

```

---

[3] // 静态数据成员 num 初始化为 0

```

int main()
{
    Test::ShowObjectNum();
    Test obj;
    Test::ShowObjectNum();

    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){ }
    Complex _____ [4] _____ (const Complex &a) const // 重载加法运算符+
    { return Complex(realPart + a.realPart, imagePart + a.imagePart); }
    void Show() const { cout << realPart << "+" << imagePart << "i" << endl; }
};

```

```

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

```

```
c = a + b;
cout << "a="; a.Show();
cout << "b="; b.Show();
cout << "c="; c.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. 设计一个类 DateInfo，要求其满足下述要求：

- （1）要求有一个无参的构造函数，其初始的年、月、日分别为：2010，6，8。
- （2）要求有一个带参数的构造函数，其参数分别对应年、月、日。
- （3）要求用一个成员函数实现日期的设置。
- （4）要求用一个成员函数实现输出日期。

2. 定义 Staff（员工）类，由 Staff 分别派生出 Saleman（销售员）类和 Manager（经理）类，要求：

（1）在 Staff 类中包含的数据成员有编号(num)、姓名(name)、出勤率(rateOfAttend)、基本工资(basicSal)和奖金(prize)。在 Saleman 类中还包含数据成员销售员提成比例(deductRate)和个人销售额(personAmount)，在 Manager 类中还包含数据成员经理提成比例(totalDeductRate)和总销售额(totalAmount)

（2）各类人员的实发工资公式如下：

员工实发工资 = 基本工资 + 奖金 \* 出勤率

销售员实发工资 = 基本工资 + 奖金 \* 出勤率 + 个人销售额 \* 销售员提成比例

经理实发工资 = 基本工资 + 奖金 \* 出勤率 + 总销售额 \* 经理提成比例

（3）每个类都有构造函数、输出基本信息函数(Output)和输出实发工资函数(OutputWage)。

## C++面向对象程序设计模拟试题五参考答案

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

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: oprator-
2. 参考答案: 继承
3. 参考答案: 私有
4. 参考答案: 重载
5. 参考答案: delete

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:  
Destructor B! 7  
Destructor A! 6  
Destructor B! 0  
Destructor A! 5
2. 参考答案: x=5, y=12
3. 参考答案: 7
4. 参考答案:  
1  
2
5. 参考答案: d=88i=88
6. 参考答案:  
0  
1  
2  
3  
4

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] this->n = n 或 Test::n = n
2. 参考答案: [2] A(x)
3. 参考答案: [3] int Test::num = 0;
4. 参考答案: [4] 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;

class DateInfo
{
private:
    int year, month, day;

public:
    DateInfo(): year(2010), month(6), day(8){ }
    DateInfo(int y, int m, int d): year(y), month(m), day(d){ }
    void Set(int y, int m, int d)
    {
        year = y;
        month = m;
        day = d;
    }
    void Show() { cout << year << "年" << month << "月" << day << "日" << endl; }
};

int main()
{
    DateInfo d1, d2(1988, 8, 18);
    d1.Show();
    d2.Show();
    d2.Set(1999, 9, 19);
    d2.Show();
    return 0;
}
```

### 2. 参考程序：

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

class Staff
{
protected:
    int num;
    char name[18];
    double rateOfAttend;
    double basicSal ;
    double prize ;
}
```



```

        static int count;

public:
    Staff(){ }
    void Input()
    {
        num = ++count;
        cout << "请输入编号为" << num << "号员工的信息" << endl;
        cout << "姓名:";
        cin >> name;
        cout << "基本工资:";
        cin >> basicSal;
        cout << "奖金:";
        cin >> prize;
        cout << "出勤率(0~1):";
        cin >> rateOfAttend;
    }
    void Output() const
    {
        cout << "编号:" << num << endl;
        cout << "姓名:" << name << endl;
        cout << "基本工资:" << basicSal << "元" << endl;
        cout << "奖金:" << prize << "元" << endl;
        cout << "出勤率:" << rateOfAttend * 100 << "%" << endl;
    }
    void OutputWage() const
    {
        cout << "实发工资:"
            << basicSal + prize * rateOfAttend << "元" << endl;
    }
};

```

```

int Staff::count = 1000;

```

```

class Saleman : public Staff

```

```

{
protected:
    float deductRate;
    float personAmount;

```

```

public:
    Saleman (){ };
    void Input()
    {

```

```

        Staff::Input();
        cout << "个人销售额:";
        cin >> personAmount;
        cout << "提成比例:";
        cin >> deductRate;
    }
    void Output() const
    {
        Staff::Output();
        cout << "个人销售额:" << personAmount << "元" << endl;
        cout << "提成比例:" << deductRate * 100 << "%" << endl;
    }
    void OutputWage() const
    {
        cout << "实发工资:"
            << basicSal + prize * rateOfAttend
            + personAmount * deductRate
            << "元" << endl;
    }
};

class Manager: public Staff
{
protected:
    double totalDeductRate;
    double totalAmount;

public:
    Manager(){ }
    void Input()
    {
        Staff::Input();
        cout << "公司总销售额:";
        cin >> totalAmount;
        cout << "提成比例:";
        cin >> totalDeductRate;
    }
    void Output() const
    {
        Staff::Output();
        cout << "公司总销售额:" << totalAmount << "元" << endl;
        cout << "提成比例:" << totalDeductRate * 100 << "%" << endl;
    }
    void OutputWage() const

```

```

    {
        cout << "实发工资:"
            << basicSal + prize * rateOfAttend
            + totalAmount * totalDeductRate
            << "元" << endl;
    }
};

int main()
{
    char flag = 'Y';

    while (toupper(flag) == 'Y')
    {
        cout << "请选择录入类别(1.员工 2.销售员 3.经理)";
        int n;
        cin >> n;

        if (n == 1)
        {
            // 员工
            Staff objStaff;
            objStaff.Input();
            objStaff.Output();
            objStaff.OutputWage();
        }
        else if (n == 2)
        {
            Saleman objSaleman;
            objSaleman.Input();
            objSaleman.Output();
            objSaleman.OutputWage();
        }
        else if (n == 3)
        {
            Manager objManager;
            objManager.Input();
            objManager.Output();
            objManager.OutputWage();
        }
        else
        {
            cout << "选择有误!" << endl;
        }
    }
}

```

```

        cout << endl << "是否继续录入信息?(Y/N)";
        cin >> flag;
    }

    return 0;
}

```

## 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题的空格内,错填或不填均无分。**

- 通过类创建(                      )来调用构造函数。
- 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(                      )。
- 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。
- 拷贝构造函数用它所在类的(                      )作为参数。
- 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;
}

```

## C++面向对象程序设计模拟试题七

**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. 下列语句中，错误的是( B )。
 

A) const int buffer = 168;
B) const int num;

C) const double \*p;
D) double \* const p = new double;
2. C++中的模板包括( c )。
 

A) 对象模板和函数模板
B) 对象模板和类模板

C) 函数模板和类模板
D) 变量模板和对象模板
3. 由于常对象不能被更新，因此 C++标准规定( A )。
 

A) 通过常对象只能调用常成员函数  
 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
4. 以下关键字不能用来声明类的访问权限的是( B )。
 

A) public
B) static
C) protected
D) private
5. 下列关于 this 指针的叙述中，正确的是( D )。
 

A) 与类相关的所有函数都含有 this 指针  
 B) 所有类的成员函数都有 this 指针  
 C) 所有函数都有 this 指针  
 D) 只有类的非静态成员函数才有 this 指针
6. 派生类的成员函数不能直接访问基类的( C )。
 

A) 保护成员
B) 公有成员

C) 私有成员
D) 前面各选项都正确
7. 在下列函数原型中，可以作为类 Test 构造函数的是( D )。
 

A) void Test (int);
B) int Test ();



- C) ~Test (int);                      D) Test (int);
8. 在下面 4 个关键字中, 用于说明虚基类的是(    A    )。
- A) virtual                              B) public
- C) protected                            D) private
9. 在 C++ 中, 用于实现编译时多态性的是(    B    )。
- A) 友元函数                            B) 重载函数
- C) 静态成员函数                        D) 虚函数
10. 下列关于 C++ 函数的说明中, 正确的是(    C    )。
- A) 内置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) C++ 标准要求在一个函数之前, 如果没定义函数, 则必须先声明其原型
- 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

## 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. 一个函数名为 Show, 返回值类型为 void, 没有参数的纯虚常成员函数可以声明为 (                      virtual void Show() const = 0;                      )。
2. 若将一个二元运算符重载为类的友元函数, 其形参个数应该是 (    2                      ) 个。
3. 采用 struct 定义的类中数据成员和成员函数的默认访问权限是 (                      公有访问 public                      )。
4. 在面向对象方法中, (    类                      ) 描述的是具有相同属性与操作的一组对象。
5. 模板类型形参表使用的关键字为(                      class                      )。

## 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 Test
{
private:
    int n;

public:
    Test(int i = 16) { n = i; }
    int Get() { return n; }
    int Get() const { return n - 2; }
};

int main()
```

```

{
    Test a;
    const Test b(18);
    cout << a.Get() << ", " << b.Get() << 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 A
{
public:
    A() { cout << "A" << endl; }
    ~A() { cout << "~A" << endl; }
};

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

int main()
{
    B obj;

    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;

int f(int a) { return a * a + 8; }

template <class T>

```

```
T f(const T &a) { return a * a; }
```

```
int main()
{
    cout << f(1) << ", " << f(1.0) << 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;
```

```
namespace ns
{
    int x = 18;
}
```

```
int x = 5;
```

```
int main()
{
    int x = 10;
    cout << x << " " << ::x << " " << ns::x << 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 A
{
public:
    virtual void Show() const { cout << "A::Show()" << endl; }
};
```

```
class B: public A
{
```

```

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

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

int main()
{
    A obj1;
    B obj2;

    Refers(obj1);
    Refers(obj2);

    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
{
protected:
    int a;

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

class B: public A
{
protected:
    int b;

public:
    B(int x, int y): A(x), b(y) {}
    void Show() const { cout << b << endl; }
}

```

```
};

int main()
{
    B obj(18, 5);
    obj.Show();
    obj.A::Show();

    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. 将如下程序补充完整。

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

class Integer
{
private:
    int n;

public:
    Integer(int n): n(n) { }
    Integer &_____ [1] _____ { --n; return *this; } // 重载前缀--运算符
    void Show() const { cout << n << endl; }
};

int main()
{
    Integer i(168);
    --i;
    i.Show();

    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 = 0): a(m) { }
    _____[2]_____ // 类型转换函数，A 类对象转成 int 型量
    { return a; }
};

```

```

int main()
{
    A a = 8.14;
    cout << a << endl;

    return 0;
}

```

3. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
private:
    int m;

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

class B: public A
{
private:
    int n;

public:
    B(int a, int b): A(a) { n = b; }
    void Show() const { cout << n << endl; }
};

```

```

int main()
{
    B obj(158, 98);
    _____[2]_____.Show();           // 调用基类的 Show()

    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){ }
    Complex _____[3]_____(const Complex &a) const           // 重载减法运算符+
    {   return Complex(realPart - a.realPart, imagePart - a.imagePart); }
    void Show() const { cout << realPart << "+" << imagePart << "i" << endl;}
};

int main()
{
    Complex a(2, 9), b(1, 1), c;

    c = a - b;
    cout << "a="; a.Show();
    cout << "b="; b.Show();
    cout << "c="; c.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积，要求编写测试程序。

函数模板声明如下：

```

template <class Type>
Type Mul(Type a[], int n);           // 求数组 a 各元素的积

```

2. 设计一个日期类 **Date**，要求：

（1）包含年(year)、月(month)和日(day)私有数据成员。

（2）包含构造函数，重载输出运算符“<<”与重载输入运算符“>>”。

要求编写测试程序。



## C++面向对象程序设计模拟试题七参考答案

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

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 void Show() const = 0;`
2. 参考答案: 2
3. 参考答案: `public` 或公有
4. 参考答案: 类
5. 参考答案: `class`

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: 16,16
2. 参考答案:  
A  
B  
~B  
~A
3. 参考答案: 9,1
4. 参考答案: 10 5 18
5. 参考答案:  
`A::Show()`  
`B::Show()`
6. 参考答案:  
5  
18

**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] operator--()
2. 参考答案: [2] operator int() const 或 operator int()
3. 参考答案: [3] obj.A::
4. 参考答案: [4] 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 Mul(Type a[], int n)                // 求数组 a 各元素的积
{
    Type t = 1;
    for (int i = 0; i < n; i++)
        t *= a[i];
    return t;
}

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

    return 0;
}
```

2. 参考程序:

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

class Date
{
private:
    int year;
    int month;
    int day;

public:
```

```

    Date(int y = 2010, int m = 1, int d = 1):year(y), month(m), day(d){ }
    void SetYear(int y) { year = y; }
    void SetMonth(int m) { month = m; }
    void SetDay(int d) { day = d; }
    int GetYear() const { return year; }
    int GetMonth() const { return month; }
    int GetDay() const { return day; }
};

istream &operator>>(istream &in, Date &dt)
{
    int y, m, d;

    cout << "输入年:";
    in >> y;
    cout << "输入月:";
    in >> m;
    cout << "输入日:";
    in >> d;
    dt.SetYear(y);
    dt.SetMonth(m);
    dt.SetDay(d);

    return in;
}

ostream &operator<<(ostream &out, const Date &dt)
{
    cout << dt.GetYear() << "年" << dt.GetMonth() << "月" << dt.GetDay() << "日";

    return out;
}

int main()
{
    Date d;
    cin >> d;
    cout << d << endl;

    return 0;
}

```

# C++面向对象程序设计模拟试题八

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. 关于函数重载，下列叙述中错误的是( )。

- A) 重载函数的函数名都相同
- B) 重载函数必须在参数个数或类型上有所不同
- C) 重载函数的返回值类型必须不相同
- D) 重载函数的函数体可以相同

2. 有以下函数模板：

```
template <class ElemType>
```

```
ElemType Square(const ElemType &x) { return x * x; }
```

其中 ElemType 是( )。

- A) 函数形参      B) 函数实参      C) 模板类型形参      D) 模板类型实参

3. 下列选项中，不属于面向对象程序设计特征的是( )。

- A) 继承性      B) 多态性
- C) 封装性      D) 类比性

4. 在 C++ 中，用于实现运行时多态性的是( )。

- A) 友元函数      B) 重载函数
- C) 静态成员函数      D) 虚函数

5. 下列关于类和对象的叙述中，错误的是( )。

- A) 一个类只能定义一个对象      B) 对象是类的具体实例
- C) 类是某一类对象的抽象      D) 类和对象的关系就像数据类型和变量的关系

6. 下面 4 个关键字中，用于说明虚函数的是( )。

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

7. 在语句 “cin >> x;” 中，cin 是( )。

- A) C++ 的关键字      B) 类名
- C) 对象名      D) 函数名

8. 运算符重载只能改变运算符原有的( )。

- A) 操作数类型      B) 操作数个数
- C) 优先级      D) 结合性

9. 为取代 C 中带参数的宏，在 C++ 中使用了( )。

- A) 重载函数      B) 内置函数
- C) 递归函数      D) 友元函数

10. 有以下类声明：

```
class MyClass{ int num; };
```

则 MyClass 类的成员 num 是( )。

- A) 公有数据成员      B) 公有成员函数      C) 私有数据成员      D) 私有成员函数

**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. 含有纯虚函数的类称为（ ）。
2. 声明静态成员使用的关键字是（ ）。
3. 一个全局函数应声明为类的（ ）函数才能访问这个类的 private 成员。
4. 重载运算符“+”的函数名为（ ）。
5. 声明全局函数为内置函数的关键字为（ ）。

**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"; }
    ~A() { cout << "~A"; }
};

int main()
{
    A obj;
    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;

class Test
{
private:
    static int count;

public:
```

```

    Test() { count++; }
    ~Test() { count--; }
    static int GetCount() { return count; }
};

int Test::count = 0;

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

template <class Type>
Type f(Type x)
{
    return x * 1.6;
}

int main()
{
    cout << f(1) << endl;
    cout << f(1.0) << endl;

    return 0;
}

```

上面程序的输出结果为：

4. 阅读下面程序，写出在 VC++ 中运行时的输出结果。

```

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

```

```

double Sqrt(double x)
{
    if (x < 0) throw "被开方数为负";
    else return sqrt(x);
}

int main()
{
    try
    {
        cout << Sqrt(9) << endl;
        cout << Sqrt(-9) << endl;
    }
    catch(char *mess)
    {
        cout << "异常信息:" << mess << 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 A
{
public:
    virtual void f() const { cout << "A::f()" << endl; }
};

class B: public A
{
public:
    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：

6. 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, 6};
    Array obj(a, 5);

    for (int i = 1; i <= 5; i++) cout << obj[i] << " ";
    cout << 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. 将如下程序补充完整。

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

class A
```



```

{
private:
    int n;

public:
    A(int n) { ____ [1] ____ = n; }           // 初始化数据成员 n
    void Show() const { cout << n << endl; }
};

```

```

int main()
{

    A i(18);
    i.Show();

    return 0;
}

```

2. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

```

```

class A
{
protected:
    int a;

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

```

```

class B: public A
{
private:
    int b;

public:
    B(int x, int y): ____ [2] ____ { b = y; }           // 将 a 初始化为 x, b 初始化为 y
    void Show() const { cout << "a:" << a << endl << "b:" << b << endl; }
};

```

```

int main()
{

```

```

        B obj(5, 18);
        obj.Show();

        return 0;
    }

```

3. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
public:
    void Show(int x = 0) const { cout << x; }
};

class B: public A
{
public:
    void Show(int x = 0) const
    {
        _____[3]_____ // 显式调用基类的函数 Show()
        cout << x << endl;
    }
};

int main()
{
    B obj;
    obj.Show(1);

    return 0;
}

```

4. 下列程序的输出结果为 1，试将程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
public:
    _____[4]_____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;
}

```

## 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和，要求编写测试程序。

函数模板声明如下：

```

template <class Type>
Type Sum(Type a[], int n);           // 求数组 a 各元素的和

```

2. 设计一个基类 Shape，Shape 中包含成员函数 Show()，将 Show()声明为纯虚函数。Shape 类公有派生矩形类 Rectangle 和圆类 Circle，分别定义 Show()实现其主要几何元素的显示。使用抽象类 Shape 类型的指针，当它指向某个派生类的对象时，就可以通过它访问该对象的虚成员函数 Show()，要求编写测试程序。

## C++面向对象程序设计模拟试题八参考答案

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

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：抽象类或抽象基类
2. 参考答案：static
3. 参考答案：友元
4. 参考答案：operator+
5. 参考答案：inline

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
2. 参考答案：  
count=1  
count=0
3. 参考答案：  
1  
1.6
4. 参考答案：  
3  
异常信息:被开方数为负
5. 参考答案：B::f()
6. 参考答案：1 2 3 4 5

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] this->n 或 Test::n
2. 参考答案：[2] A(x)

3. 参考答案: [3] A::Show();

4. 参考答案: [4] void

## 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 Sum(Type a[], int n)           // 求数组 a 各元素的和
{
    Type s = 0;
    for (int i = 0; i < n; i++)
        s += a[i];
    return s;
}

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

    return 0;
}
```

2. 参考程序:

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

const double PI = 3.1415926;

class Shape
{
public:
    virtual ~Shape() { }
    virtual void Show() const = 0;
};

class Rectangle: public Shape
{
private:
    double height;
    double width;
```

```

public:
    Rectangle(double h, double w): height(h), width(w) { }
    void Show() const
    {
        cout << "矩形:" << endl;
        cout << "高:" << height << endl;
        cout << "宽:" << width << endl;
        cout << "周长:" << 2 * (height + width) << endl;
        cout << "面积:" << height * width << endl << endl;
    }
};

class Circle: public Shape
{
private:
    double radius;

public:
    Circle(double r): radius(r) { }
    void Show() const
    {
        cout << "圆形:" << endl;
        cout << "半径:" << radius << endl;
        cout << "周长:" << 2 * PI * radius << endl;
        cout << "面积:" << PI * radius * radius << endl << endl;
    }
};

int main()
{
    Shape *p;
    p = new Circle(1);
    p->Show();
    delete p;
    p = new Rectangle(1, 2);
    p->Show();
    delete p;
    return 0;
}

```

## C++面向对象程序设计模拟试题九

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. C++中的模板包括（ ）。

- A) 对象模板和函数模板                      B) 对象模板和类模板  
C) 函数模板和类模板                      D) 变量模板和对象模板

3. 有以下类声明：

```
class MyClass{ int num; };
```

则 MyClass 类的成员 num 是（ ）。

- A) 公有数据成员                      B) 公有成员函数  
C) 私有数据成员                      D) 私有成员函数

4. 在 C++中，用于实现静态多态性的是( )。

- A) 友元函数                      B) 重载函数  
C) 静态成员函数                      D) 虚函数

5. 派生类的成员函数不能访问基类的( )。

- A) 公有成员和保护成员                      B) 公有成员  
C) 私有成员                      D) 保护成员

6. 在语句“cin >> data;”中，cin 是( )。

- A) C++的关键字                      B) 类名                      C) 对象名                      D) 函数名

7. 下列关于输入流类成员函数 getline()的描述中，错误的是( )。

- A) 该函数可以用来读取键盘输入的字符串  
B) 该函数读取的字符串长度不受任何限制  
C) 该函数读取字符串时，遇到换行符便停止  
D) 该函数读取字符串时，可以包含空格

8. 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) 标准 C++要求在调用一个函数之前, 如果没定义函数, 则必须先声明其原型

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

9. 为取代 C 中带参数的宏, 在 C++中使用了( )。

A) 重载函数      B) 内联函数      C) 递归函数      D) 友元函数

10. 声明运算符重载的关键字的是( )。

A) void      B) int      C) operator      D) double

## 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. 一个函数名为 GetArea, 返回值类型为 double, 没有参数的纯虚常成员函数可以声明为\_\_\_\_\_。

2. 在基本类型与类类型的转换中, 类类型转换函数的作用是\_\_\_\_\_。

3. 派生类中的成员函数可以直接访问基类中的\_\_\_\_\_成员。

4. 重载函数运算符的函数名为\_\_\_\_\_。

5. 在 C++流类库中, 根基类为\_\_\_\_\_。

## 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 Int
```

```
{
```

```
private:
```

```
    int n;
```

```
public:
```

```
    Int(int i = 9): n(i) { }
```

```
    int Get() { return n; }
```

```
    int Get() const { return n - 1; }
```

```
    operator int() const { return n; }
```



```
};
```

```
int main()
```

```
{
```

```
    Int a;
```

```
    const Int b = 8;
```

```
    cout << a.Get() << "," << b.Get() << "," << a << "," << b << "," << "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 A
```

```
{
```

```
public:
```

```
    A() { cout << "A()" << endl; }
```

```
    ~A() { cout << "~A()" << endl; }
```

```
};
```

```
int main()
```

```
{
```

```
    A *p = new A, a;
```

```
    delete p;
```

```
    cout << "end" << 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;

template <class Type>
Type f(const Type &x) { return x + 1; }

int f(int x) { return x + 2.1; }

int f(double x) { return x + 3; }

int main()
{
    cout << f(96) << endl;
    cout << f(96.0) << endl;
    cout << f(float(96.0)) << endl;
    cout << f(char(96.0)) << endl;
    cout << f('a') << 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;

int x = 1;
```

```

namespace ns
{
    int x = 6;

    int f(int x) { return ::x + 1; }

    int f(double x) { return x + 1; }
}

int main()
{
    int x = 8;

    cout << x << " " << ::x << " " << ns::x << " " << ns::f(1) << " " << ns::f(1.0) << 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 A
{
public:
    virtual void f() const { cout << "A::f()" << endl; }
};

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

```

```
void Refer(const A &a) { a.f(); }
```

```
int main()
```

```
{
```

```
    A a, *p;
```

```
    B b;
```

```
    Refer(a);
```

```
    Refer(b);
```

```
    p = &a;
```

```
    p->f();
```

```
    p = &b;
```

```
    p->f();
```

```
    p->A::f();
```

```
    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;
```

```
template <class Type, int size>
```

```
class Array
```

```
{
```

```
private:
```

```
    Type elem[size];
```

```
public:
    Array(Type a[]) { for (int i = 0; i < size; i++) elem[i] = a[i]; }
    int Length() const { return size; }
    Type &operator[](int i){ return elem[i - 1]; }
};
```

```
int main()
{
    int a[] = { 1, 2, 3, 4, 5,};
    const int n = 5;
    Array<int, n> obj(a);

    for (int i = 1; i <= obj.Length(); i++) cout << obj[i] << " ";
    cout << endl;

    return 0;
}
```

上面程序的输出结果为：

#### 四、程序填空题（本大题共 4 小题，每小题 3 分，共 12 分）

1. 将如下程序补充完整。

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

class Double
{
private:
    double num;

public:
```

```

    Double(double num) { _____ = num; } //初始化数据成员 num 为形参 num

    void Show() const { cout << num << endl; }

};

int main()
{

    Double x(1.8);

    x.Show();

    return 0;

}

```

2. 将如下程序补充完整。

```

#include <iostream>

using namespace std;

class Int
{
private:
    int n;

public:
    Int(int m): n(m) { }

    Int _____{ ++n; return *this; }// 重载前缀++运算符

    void Show() const { cout << n << endl; }

};

int main()
{

```

```

    Int i(8);

    (i++).Show();

    i.Show();

    return 0;
}

```

3. 将如下程序补充完整。

```

#include <iostream>

using namespace std;

class A
{
protected:
    int m;

public:
    A(int a): _____{ }           // 用 a 初始化 m
    void Show() const { cout << m << endl; }
};

int main()
{
    A obj(1);
    obj.Show();

    return 0;
}

```

4. 将如下程序补充完整。

```

#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; }
};

_____ ;           // 静态数据成员的初始化

int main()
{
    Point::ShowCount();           // 输出 0
    Point a(6, 8);
    Point::ShowCount();           // 输出 1

    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大值，要求编写测试程序。



函数模板声明如下：

2. 编写程序实现如下功能：

（1）从键盘上输入一系列学生数据(学号、姓名、成绩)，并将这些学生数据写入到文件 `stud.dat` 中。

（2）显示文件 `stu.dat` 中的学生数据和相应的成绩等级( $\geq 90$  为优，80~89 为良，70~ 79 中，60~69 为及格， $\leq 59$  为不及格)。

## C++面向对象程序设计模拟试题九参考答案

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. C | 3. C | 4. B | 5. C  |
| 6. C | 7. B | 8. C | 9. B | 10. C |

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 double GetArea() const = 0;`
2. 参考答案: 将类类型转换为基本类型
3. 参考答案: 公有与保护
4. 参考答案: `operator()`
5. 参考答案: `ios`

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:  
9,7,9,8,end

2. 参考答案:  
A()  
A()  
~A()  
end  
~A()

3. 参考答案:  
98  
99

97

a

b

4. 参考答案:

8 1 6 2 2

5. 参考答案:

A::f()

B::f()

A::f()

B::f()

A::f()

6. 参考答案:

1 2 3 4 5

#### 四、程序填空题（本大题共 4 小题，每小题 3 分，共 12 分）

1. 参考答案: this->num 或 Int:: num

2. 参考答案: operator++()

3. 参考答案: m(a)

4. 参考答案: int Point::count = 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. 参考程序:

```
#include <iostream> // 编译预处理命令
```

```
#include <cmath> // 编译预处理命令
```

```

using namespace std;                // 使用命名空间 std

template <class Type>
Type Max(Type a[], int n)           // 求数组 a 各元素的最大值
{
    Type m = a[0];                  // 定义变量 m, 用于表示当前所求的最大值
    for (int i = 1; i < n; i++) m = m < a[i] ? a[i]: m;    // 依次比较求最大值
    return m;                        // 返回最大值
}

int main()                          // 主函数 main()
{
    double a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9};    // 定义数组
    cout << Max(a, 9) << endl;    // 输出各元素的最大值

    return 0;                        // 返回值, 返回操作系统
}

```

## 2. 参考程序:

```

#include <iostream>                  // 预处理命令
#include <fstream>                    // 预处理命令
#include <iomanip>                    // 流格式控制预处理命令
using namespace std;                // 使用标准命名空间 std

// 学生类
class Student
{
private:
// 数据成员:
    int no;                          // 学号
    char name[16];                    // 姓名
    int score;                        // 成绩

```

```

public:
// 公有函数:

void Input()                // 输入信息

{ cin >> no >> name >> score; }

void Show() const           // 显示信息

{

    char level[6];          // 等级

    if (score >= 90) strcpy(level, "优");        // >=90 为优

    else if (score >= 80) strcpy(level, "良");    // 80~89 为良

    else if (score >= 70) strcpy(level, "中");    // 70~79 为中

    else if (score >= 60) strcpy(level, "及格");  // 60~69 为及格

    else strcpy(level, "不及格");                // <60 为不及格

    cout << setw(16) << no << setw(18) << name << setw(16) << score << setw(16) <<
level << endl;

}

};

// 学生管理类

class StudentManage

{

private:

// 辅助函数:

    static void Write();        // 写入学生信息

    static void Show();         // 显示学生信息

public:

// 公有函数:

    static void Run();          // 运行学生管理

};

// 学生管理类的实现部分

```

```

void StudentManage::Write()           // 写入学生信息
{
    Student objStu;                   // 学生对象
    ofstream outFile("stud.dat", ios::binary | ios::app); // 定义文件对象
    if (!outFile)
    { // 打开文件失败
        cout << "打开文件失败!" << endl;
        exit(1);
    }

    char flag = 'Y';                  // 判断是否继续录入的标志, 初始化为'Y'
    cout << "输入学生信息数据(学号 姓名 成绩):" << endl;
    while (toupper(flag) == 'Y')
    {
        objStu.Input();               // 输入学生信息
        outFile.write((char *)&objStu, sizeof(objStu)); // 写入文件
        cout << "是否继续录入信息?(Y/N)";
        cin >> flag;
    }

    outFile.close();                  // 关闭文件
}

void StudentManage::Show()           // 显示学生信息
{
    Student objStu;                   // 学生对象
    ifstream inFile("stud.dat", ios::binary); // 定义文件对象
    if (!inFile)
    { // 打开文件失败
        cout << "打开文件失败!" << endl;
        exit(2);
    }
}

```

```

    cout << "显示学生信息" << endl;

    cout << setw(16) << "学号" << setw(18) << "姓名" << setw(16) << "成绩" << setw(16)
    << "等级" << endl;

    inFile.read((char *)&objStu, sizeof(objStu));    // 从文件读数据
    while (!inFile.eof())
    {    // 循环显示学生信息
        objStu.Show();            // 显示学生信息
        inFile.read((char *)&objStu, sizeof(objStu)); // 从文件读数据
    }

    inFile.close();                // 关闭文件
}

void StudentManage::Run()        // 运行学生管理
{
    int select;                    // 临时变量

    do
    {
        cout << "选择(1.写入学生信息 2.显示学生信息 3.退出):";
        cin >> select;            // 输入选择
        switch (select)
        {
            case 1:
                Write();           // 写入学生信息
                break;
            case 2:
                Show();            // 显示学生信息
                break;
            case 3:
                break;             // 退出
        }
    }
}

```

```

        default:                // 其它情况, 表示选择有误

            cout << "选择有误!"<< endl;

            break;

    }

} while (select != 3);

}

int main(void)                  // 主函数 main(void)
{

    StudentManage::Run();       // 运行学生管理

    system("PAUSE");            // 调用库函数 system()

    return 0;                   // 返回值 0, 返回操作系统

}

```

## C++面向对象程序设计模拟试题十

**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. 说明虚基类的关键字是（ ）。

A) inline                  B) virtual                  C) public                  D) static

2. 有以下函数模板：

```
template <class ElemType>
```

```
ElemType Square(const ElemType &x) { return x * x; }
```

其中 ElemType 是（ ）。

A) 函数形参                  B) 函数实参                  C) 模板类型形参                  D) 模板类型实参

3. 在下列函数原型中，可以作为类 AA 构造函数的是（ ）。

A) void AA(int);                  B) int AA();                  C) AA(int) const                  D) AA(int);

4. 下列关于类和对象的叙述中，**错误**的是（ ）。

A) 一个类只能有一个对象                  B) 对象是类的具体实例



C) 类是某一类对象的抽象                      D) 类和对象的关系就像数据类型和变量的关系

5. 下列有关继承和派生的叙述中, 正确的是 (            )。

A) 如果一个派生类私有继承其基类, 则该派生类不能访问基类的保护成员

B) 派生类的成员函数可以访问基类的所有成员

C) 基类对象可以赋值给派生类对象

D) 派生类对象可以赋值给基类对象

6. 语句 `ofstream f ("test.txt", ios::out)` 的功能是建立与流对象 `f` 的关联, 而且 (            )。

A) 若文件存在, 将其置为空文件; 若文件不存在, 打开失败

B) 若文件存在, 将文件指针定位于文件尾; 若文件不存在, 建立一个新文件

C) 若文件存在, 将清除文件; 若文件不存在, 建立一个新文件

D) 若文件存在, 打开失败, 若文件不存在, 建立一个新文件

7. 在 C++ 中, 用于实现运行时多态性的是 (            )。

A) 友元函数            B) 重载函数            C) 模板函数            D) 虚函数

8. 下列情况中, **不会**调用复制构造函数的是 (            )。

A) 用一个对象去初始化同一类的另一个新对象时

B) 将类的一个对象赋予该类的另一个对象时

C) 函数的形参是类的对象, 调用函数进行形参和实参结合时

D) 函数的返回值是类的对象的引用, 函数执行返回调用时

9. 下列有关内置函数的叙述中, 正确的是 (            )。

A) 内置函数在调用时发生控制转移

B) 内置函数必须通过关键字 `inline` 来定义

C) 内置函数是通过编译器来实现的

D) 内置函数体的最后一条语句必须是 `return` 语句

10. 声明类类型转换函数的关键字的是 (            )。

A) `void`            B) `int`            C) `operator`            D) `double`

## 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. 一个函数名为 `Show`, 返回值类型为 `void`, 没有参数的纯虚常成员函数可以声明为\_\_\_\_\_。

2. 在基本类型与类类型的转换中, 转换构造函数的作用是\_\_\_\_\_。

3. 派生类中的成员函数**不能**直接访问基类中的\_\_\_\_\_成员。

4. 重载下标运算符的函数名为\_\_\_\_\_。
5. 在 C++ 流类库中, istream 和 ostream 的基类为\_\_\_\_\_。

### 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; }
};

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

int main()
{
    B obj;

    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 A
{
public:
    A() { count++; }
    ~A() { count--; }
    static int GetCount() { return count; }
private:
    static int count;
};

int A::count = 0;

int main()
{
    cout << A::GetCount() << endl;
    A a, b;
    cout << A::GetCount() << endl;
    A *p = new A[9], *q = new A;
    cout << A::GetCount() << endl;
    delete []p;
    cout << A::GetCount() << endl;
    delete q;
    cout << A::GetCount() << 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;

template <class Type>
Type Fun(const Type &x) {    return x + 1; }

int Fun(int x) { return x + 2; }

double Fun(double x) { return x + 3; }

int main()
{
    cout << Fun(6) << endl;
    cout << Fun(6.0) << endl;
    cout << Fun(float(6)) << endl;
    cout << Fun(double(6)) << endl;
    cout << Fun('a') << 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;

namespace
{
```

```

    int x = 1;
}

namespace ns
{
    int x = 6;

    int f(int x) { return x + 1; }

    int f(double x) { return x + 2; }
}

int main()
{
    cout << x << " " << ns::x << " " << ns::f(1) << " " << ns::f(1.0) << " end" << 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 A
{
public:
    virtual void Fun() const { cout << "A" << endl; }
};

class B: public A
{
public:

```

```
void Fun() const { cout << "B" << endl; }  
};
```

```
void Refer(const A &a) { a.Fun(); }
```

```
int main()  
{  
    A a, *p = &a;  
    B b;  
  
    Refer(a);  
    Refer(b);  
    p->Fun();  
    p = &b;  
    p->Fun();  
    p->A::Fun();  
  
    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;
```

```
template <class Type>  
class Array  
{  
private:  
    Type *elem;  
    int size;
```

```

public:
    Array(Type a[], int n): size(n)
    {
        elem = new Type[size];
        for (int i = 0; i < size; i++) elem[i] = a[i];
    }
    ~Array() { delete []elem; }
    int Size() const { return size; }
    Type &operator[](int i){ return elem[i - 1]; }
};

int main()
{
    int a[] = { 1, 2, 3, 4, 5 }, n = 5;
    Array<int> obj(a, n);

    for (int i = 1; i <= obj.Size(); i++) cout << obj[i] << " ";
    cout << endl;

    return 0;
}

```

上面程序的输出结果为：

#### 四、程序填空题（本大题共 4 小题，每小题 3 分，共 12 分）

1. 将如下程序补充完整。

```

#include <iostream>

using namespace std;

class Double

```

```

{
private:
    double num;

public:
    Double(double n) : _____ { } //初始化数据成员 num 为形参 n
    void Show() const { cout << num << endl; }
};

int main()
{

    Double x(1.8);
    x.Show();

    return 0;
}

```

参考答案：num(n)

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; }
    virtual void Show() const { cout << a << endl; }
}

```



```
};
```

```
class B: public A
```

```
{
```

```
private:
```

```
    int b;
```

```
public:
```

```
    B(int m, int n): A(m), b(n) { }
```

```
    void Show() const { _____; cout << b << endl; }// 分行显示 a 与 b 之值  
};
```

```
int main()
```

```
{
```

```
    B b(1, 6);
```

```
    A *p = &b;
```

```
    p->Show();
```

```
    return 0;
```

```
}
```

3. 将如下程序补充完整。

```
#include <iostream>
```

```
using namespace std;
```

```
class A
```

```
{
```

```
protected:
```

```
    int a;
```

```
public:
```

```

A(int a) { _____; }           // 用形参 a 初始化数据成员 a
void Show() const { cout << a << endl; }
};

```

```

int main()
{
    A obj(1);
    obj.Show();

    return 0;
}

```

4. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class Int
{
private:
    int n;

public:
    Int(int m): n(m) { }
    Int _____(const Int &a) { return Int(n + a.n); } // 重载加法运算符
    void Show() const { cout << n << endl; }
};

int main()
{

    Int a(3), b(6), c = a + b;
}

```

```
c.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输出一个数组各元素的值，要求编写测试程序

函数模板声明如下：

```
template <class Type>  
void Show(Type a[], int n);           // 输出一个数组各元素的值
```

2. 编写程序实现如下功能：

（1）从键盘上输入一系列员工工资信息(姓名、工资)，并将这些员工工资信息写入到文件 `employee.dat` 中。

（2）显示文件 `employee.dat` 中的员工工资信息和所有员工的平均工资。

## C++面向对象程序设计模拟试题十参考答案

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

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 void Show () const = 0;`
2. 参考答案: 将基本类型转换为类类型
3. 参考答案: 私有或 `private`
4. 参考答案: `operator[]`
5. 参考答案: `ios`

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
B
end
~B
~A
```

2. 参考答案:

```
0
2
12
3
```

2

3. 参考答案:

8

9

7

9

b

4. 参考答案:

1 6 2 3 end

5. 参考答案:

A

B

A

B

A

6. 参考答案:

1 2 3 4 5

#### 四、程序填空题（本大题共 4 小题，每小题 3 分，共 12 分）

1. 参考答案: num(n)

2. 参考答案: A::Show()

3. 参考答案: this->a = a 或 A:: a = a

4. 参考答案: 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>                // 编译预处理命令
using namespace std;              // 使用命名空间 std

template <class Type>
void Show(Type a[], int n)        // 输出一个数组各元素的值
{
    for (int i = 0; i < n; i++) cout << a[i] << " ";    // 依次输出各元素之值
    cout << endl;                // 换行
}

int main()                        // 主函数 main()
{
    double a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9}; // 定义数组
    Show(a, 9);                    // 输出一个数组各元素的值

    return 0;                      // 返回值, 返回操作系统
}
```

2. 参考程序:

```
#include <iostream>                // 编译预处理命令
#include <fstream>                  // 编译预处理命令
#include <iomanip>                  // 编译预处理命令
using namespace std;              // 使用命名空间 std

struct Employee                   // 员工结构
{
    char name[16];                // 姓名
```

```

float wage;                // 工资
};

int main()                  // 主函数 main()
{
    Employee e;            // 定义员工对象
    char answer;           // 定义字符变量
    float average = 0;     // 平均工资
    int n = 0;             // 人数

    fstream f;             // 定义文件对象

    f.open("employee.dat", ios::out | ios::binary); // 打开文件
    if (f.fail())
    { // 打开文件失败
        cout << "打开文件失败!" << endl;
        exit(1);           // 退出程序
    }
    do
    {
        cout << "输入姓名与工资:"; // 输入提示
        cin >> e.name >> e.wage;   // 输入姓名与工资
        f.write((char *)&e, sizeof(Employee)); // 写数据到文件中
        cout << "是否继续输入(y/n)?"; // 输入提示
        cin >> answer;             // 输入选择
        answer = tolower(answer);  // 转换小写字母
        while (answer != 'y' && answer != 'n')
        {
            cout << "应输入(y/n)?"; // 输入提示
            cin >> answer;           // 输入选择
            answer = tolower(answer); // 转换小写字母
        }
    }
}

```

```

    } while (answer == 'y');

    f.close();                                // 关闭文件

    f.open("employee.dat", ios::in | ios::binary);    //打开文件
    if (f.fail())
    {    // 打开文件失败
        cout << "打开文件失败!" << endl;
        exit(1);                                // 退出程序
    }

    cout << "员工信息" << endl;                // 输出提示
    cout << setw(16) << "姓名" << setw(21) << "工资" << endl;
    f.seekg(0);                                // 重新定位于文件开始处
    f.read((char *)&e, sizeof(Employee)); // 读出第 1 个学生的信息
    while (!f.eof())
    {    // 文件未结束
        cout << setw(16) << e.name << setw(19) << e.wage << "元" << endl; // 显示信息
        average = average + e.wage;    // 累加求和
        n++;                            // 统计人数
        f.read((char *)&e, sizeof(Employee));    // 读出员工的信息
    }

    if (n > 0)
        cout << "平均工资:" << average / n << "元" << endl;    // 显示信息
    f.close();                                // 关闭文件

    return 0;                                // 返回值 0, 返回操作系统
}

```