

计算机学院 2015 级高级语言程序设计 II 计期末试题 A 卷

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

参考答案：A

2. 有以下函数模板声明：

```
template <class Type>
Type Fun(const Type &x, const Type &y) { return x * 6 + y * 18; }
```

在下列对 Fun() 的调用中，**错误**的是（ ）。

- A) Fun(5, 16); B) Fun(3.1, 8.06); \
- C) Fun(6, 5.0); D) Fun (6, int(5.19));

参考答案：C

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) 泛型性

参考答案：D

4. 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

参考答案：C

5. 用来派生新类的类称为()。

- A) 新生类 B) 派生类 C) 父类 D) 子类

参考答案：C

6. 下列不具有访问权限属性的是（ ）。

- A. 非类成员 B. 类成员 C. 数据成员 D. 函数成员

参考答案：A

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

参考答案：B

8. 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

参考答案：B

9. 有以下类声明：

```
struct CTest{ float f; };
```

则类 CTest 的成员 f 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

参考答案: A

10. 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

参考答案: 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. 一个函数名为 Area，返回值类型为 double，无参数的纯虚常成员函数可以声明为_____。

参考答案: virtual double Area() const = 0;

2. 若将一个二目运算符重载为全局函数或类的友元函数，其形参个数应该是_____个。

参考答案: 2

3. 在基本类型与类类型的转换中，类类型转换函数的作用是_____。

参考答案: 将类类型转换为基本类型

4. 用关键字 static 修饰的成员称为_____。

参考答案: 静态成员

5. 重载运算符 “*” 的函数名为 ()。

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. 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 = 8): n(i) { }
    int Get() { return n; }
    int Get() const { return n + 2; }
    operator int() const { return n + 3; }
};

int main()
{
    Int a;
    const Int b = 6;
    cout << a.Get() << ", " << b.Get() << ", " << a << ", " << b << ", " << "end" << endl;

    return 0;
}
```

```
}
```

上面程序的输出结果为：

参考答案：

8,8,11,9,end

2. 阅读下面程序，写出输出结果。

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

class Test
{
public:
    static int count;
    Test() { count++; }
    ~Test() { count--; }
};

int Test::count = 201600;

int main()
{
    cout << Test::count << endl;
    Test obj;
    cout << Test::count << endl;
    Test *p = new Test[18], *q = new Test;
    cout << Test::count << endl;
    delete []p;
    cout << Test::count << endl;
    delete q;
    cout << Test::count << endl;

    return 0;
}
```

上面程序的输出结果为：

参考答案：

201600

201601

201620

201602

201601

3. 阅读下面程序，写出输出结果。

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

template <class ElemType>
ElemType f(ElemType x)
{
    ElemType y;
    y = x * (ElemType)5.8;
    return y;
}
```

```

    }

    int main()
    {
        cout << f(5) << " " << f(5.0) << " " << f(1.1) << " " << f((int)1.1) << " end" << endl;

        return 0;
    }

```

上面程序的输出结果为：

参考答案：

25 29 6.38 5 end

4. 阅读下面程序，写出输出结果。

```

#include <iostream>
using namespace std;

template <class Type>
class CTest
{
public:
    CTest(Type tA = 0, Type tB = 0):m_tA(tA), m_tB(tB){ }

    void Show()
    {
        cout << m_tA << endl;
        cout << m_tB << endl;
    }

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

private:
    Type m_tA;
    const Type m_tB;
};

int main(void)
{
    CTest<float> oTest1;
    oTest1.Show();

    CTest<int> oTest2(1, 8);
    oTest2.Show();

    const CTest<double> oTest3(6, 8);
    oTest3.Show();

    return 0;
}

```

上面程序的输出结果为：

参考答案：

0
0
1
8
8

5. 阅读下面程序，写出输出结果。

```
#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(1) << endl;
        cout << Sqrt(4) << endl;
        cout << Sqrt(9) << endl;
        cout << Sqrt(0) << endl;
        cout << Sqrt(-1) << endl;
        cout << Sqrt(-4) << 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：

参考答案：

1
2
3
0
异常信息:被开方数为负

6. 阅读下面程序，写出输出结果。

```
#include <iostream>
```

```

using namespace std;

class Base
{
public:
    virtual void f() { cout << "Base::f()" << endl; }
};

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

int main()
{
    Derived obj;
    Base *p = &obj;
    p->f();
    p->Base::f();
    p = new Base;
    p->f();
    p->Base::f();
    cout << "end" << endl;

    return 0;
}

```

上面程序的输出结果为：

参考答案：

Derived::f()

Base::f()

Base::f()

Base::f()

end

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 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;
}

```

参考答案： operator++()

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){ }
    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
    {
        _____;           // 调用基类的成员函数 Show()
        cout << n << endl;
    }
};

int main()
{
    B obj(1, 8);
    obj.Show();

    return 0;
}

```

参考答案: A::Show();

4. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

struct A
{
    _____Show() const { cout << 1 << endl; }
};

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

int main()
{
    A *p = new B;
    p->Show();           // 输出 2
    delete p;

    return 0;
}

```

参考答案: virtual 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. 试使用函数模板实现求一个数组各元素的平方值之和，要求编写测试程序。

函数模板声明如下：

```
template <class Type>
Type SumOfSquare(Type a[], int n);           // 求数组 a 各元素的最大值

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 SumOfSquare(Type a[], int n)           // 求数组 a 各元素的最大值
{
    Type s = a[0];                          // 定义变量 m, 用于表示和
    for (int i = 1; i < n; i++)             // 依次累加各元素的平方值
        s += a[i] + a[i];                  // 返回平方值的和
    return s;
}

int main()                                  // 主函数 main()
{
    double a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9}; // 定义数组
    cout << SumOfSquare(a, 9) << endl;       // 输出各元素的平方值之和

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

2. 定义抽象基类 Shape(形状)，由它派生出 2 个派生类: Circle(圆形)和 Rectangle(矩形)，用函数 GetArea()返回各种图形的面积，用函数 GetShapeName()返回图形名。在类 Shape(形状)中将函数 GetArea()与 GetShapeName()声明为纯虚函数。它们的具体定义放在派生类中实现，最后还要显示所有图形的总面积，要求编写测试程序。

参考程序：

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

const double PI = 3.1415926;                // 常量 PI

// 声明形状抽象类
class Shape
{
public:
    // 公有成员:
    virtual ~Shape() { }                    // 虚析构函数
    virtual double GetArea() const = 0;      // 纯虚函数, 返回面积
    virtual char *GetShapeName() const = 0;  // 纯虚函数, 返回图形名称
    static double sum;                       // 静态数据成员
};

// 声明圆形类
```

```

class Circle: public Shape
{
private:
// 数据成员:
    double radius;                                // 半径

public:
// 公有函数:
    Circle(double r): radius(r)                    // 构造函数
    { sum += PI * radius * radius; }
    double GetArea() const { return PI * radius * radius; } // 返回圆面积
    virtual char *GetShapeName() const { return "圆"; }     // 返回图形名称
};

// 声明矩形类
class Rectangle: public Shape
{
private:
// 数据成员:
    double height;                                // 高
    double width;                                 // 宽

public:
// 公有函数:
    Rectangle(double h, double w): height(h), width(w)      // 构造函数
    { sum += height * width; }
    double GetArea() const { return height * width; }         // 返回矩形面积
    virtual char *GetShapeName() const { return "矩形"; }     // 返回图形名称
};

double Shape::sum = 0;                                // 为静态数据成员赋初值

int main()                                             // 主函数 main()
{
    char flag = 'Y';                                  //继续录入的标志, 初始化为'Y'
    Shape *p;                                         // 基类指向

    while (toupper(flag) == 'Y')
    {
        cout << "请选择输入类别(1.圆形 2.矩形)";
        int select;                                  // 临时变量
        cin >> select;                                // 输入选择

        switch (select)
        {
        case 1:                                       // 圆形
            double r;                                // 半径
            cout << "输入半径:";
            cin >> r;                                  // 输入半径
            p = new Circle(r);                        // 生成圆对象
            cout << p->GetShapeName() << "的面积为" << p->GetArea() << endl;
            // 输出相关信息
            delete p;                                  // 释放存储空间
            break;
        case 2:                                       // 矩形
            double h, w;                              // 高宽
            cout << "输入高:";
            cin >> h;                                  // 输入高

```

```

        cout << "输入宽:";
        cin >> w;                                // 输入宽
        p = new Rectangle(h, w);                  // 生成矩形对象
        cout << p->GetShapeName() << "的面积为" << p->GetArea() << endl;
        // 输出相关信息
        delete p;                                  // 释放存储空间
        break;
    default:                                       // 其它情况, 表示选择有误
        cout << "选择有误!" << endl;
        break;
    }

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

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

```

四川大学期末考试试题（闭卷）

（2016——2017 学年第 2 学期）A 卷

课程号: 304024030 课序号: 课程名称: 高级语言程序设计 II 任课教师: 成绩:

适用专业年级: 计算机专业 16 级 学生人数: 印题份数: 学号: 姓名:

考 生 承 诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺：

- 1、已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；
- 2、不带手机进入考场；
- 3、考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。

考生签名：

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（ c ）。
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（ D ）。
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 ）。
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（ D ）。
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（ C ）。
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（ D ）。
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（ D ）。
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（ C ）。
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（ C ）。
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 virtual void show() const = 0;。

2. 在基本类型与类类型的转换中，转换构造函数的作用是将基本类型转化为类类型。

3. 派生类中的成员函数不能直接访问基类中的 private 成员。

4. 重载下标运算符的函数名为 operator[]()。

5. 在 C++ 流类库中，istream 和 ostream 的基类为 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. 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;
}

```

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。

试
卷
编
号：

四川大学期末考试试题（闭卷）

（2017——2018 学年第 2 学期）A 卷

课程号：304024030 课序号： 课程名称：高级语言程序设计 II 任课教师： 成绩：
适用专业年级：计算机专业 16 级 学生人数： 印题份数： 学号： 姓名：

考 生 承 诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺：

- 1、已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；
- 2、不带手机进入考场；
- 3、考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。

考生签名：

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（ B ）。
A) private B) virtual C) public D) static
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. 在下列函数原型中，可以作为类 CTest 的析构函数的是（ D ）。
A) void ~CTest(int); B) void ~CTest(); C) ~CTest(int) D) ~CTest();
4. 有以下类声明：
class MyClass{ int num; };
则 MyClass 类的成员 num 是（ C ）。
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
5. 对于语句“cout << endl << x;”中的各个组成部分，下列叙述中**错误**的是（ D ）。
A) “cout” 是一个输出流对象 B) “endl” 的作用是输出回车换行
C) x 是一个标识符 D) “<<” 称为输入运算符
6. 在 C++中，用于实现**运行时多态性**的是（ B D ）。
A) 友元函数 B) 重载函数 C) 模板函数 D) 虚函数
7. 下列有关**继承和派生**的叙述中，**正确**的是（ D ）。
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
8. 由于**常对象不能被更新**，因此（ A ）。
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
9. 运算符重载是对已有的运算符赋予多重含义，因此（ C ）。
A) 可以对基本类型（如 int 类型）的数据，重新定义 “+” 运算符的含义
B) 可以改变一个已有运算符的优先级和操作数个数
C) 只能重载 C++中已经有的运算符，不能定义新运算符
D) C++中已经有的所有运算符都可以重载
10. 声明运算符重载的关键字的是（ C ）。
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. 非成员函数应声明为类的_____友元_____函数才能访问这个类的 private 成员。

2. 已知 `int dbl(int n) { return n + n; }` 和 `long dbl(long n) { return n + n; }` 是一个函数模板的两个实例，则该函数模板的定义是

```
template <typename ElemType>
_____ ElemType db(ElemType n) {return n + n;} _____
```

3. 派生类中的成员函数能够直接访问基类中的_____public 与 protected_____成员。

4. 重载函数运算符的函数名为_____operator()_____。

5. 由于 ios 是_____抽象基类_____类，所以不能用 ios 直接定义对象。

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 << "1 "; }
```

```
    ~A() { cout << "2  "; }  
};
```

```
class B: public A  
{  
public:  
    B() { cout << "3  "; }  
    ~B() { cout << "4  "; }  
};
```

```
int main()  
{  
    B obj;  
    cout << "end  ";  
  
    return 0;  
}
```

上面程序的输出结果为：1， 3， 4， 2

2. 阅读下面程序，写出输出结果。

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

```
class MyClass  
{  
public:  
    MyClass() { cout << "1  "; }  
    ~MyClass() { cout << "2  "; }  
};
```

```
int main()  
{  
    MyClass *p = new MyClass;  
    delete p;  
    MyClass a;
```

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

```
return 0;
```

```
}
```

上面程序的输出结果为：1， 2， 1， 2

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; }
```

```
int Fun(int x) { return x * x; }
```

```
double Fun(double x) { return x * x * x; }
```

```
int main()
```

```
{
```

```
    cout << Fun(2) << " " << Fun(2.0) << " " << Fun(float(2))  
        << " " << Fun(double(2)) << " " << Fun('x');
```

```
    return 0;
```

```
}
```

上面程序的输出结果为：4， 8， 2， 8， x

4. 阅读下面程序，写出输出结果。

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

namespace MyName
{
    int x = 1;
}

int x = 2;

int main(void)
{
    int x = 3;

    cout << MyName::x << " " << ::x << " " << x;
    namespace My = MyName;
    cout << " " << My::x << " end";

    return 0;
}
```

上面程序的输出结果为：1, 2 3 1

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 << "1" << endl; }
        void Show() { cout << "2" << endl; }
};

class B: public A
{
public:
    void Show() const { cout << "3" << endl; }
};

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

int main()
{
    A a, *p = &a;
    const A b;
    B c;

    Refer(a);
    Refer(b);
    p->Show();
    p = &c;
    p->Show();
    b.Show();

    return 0;
}

```

上面程序的输出结果为: **2 1 2 3 1 -》 1 1 2 2 (根本就没有调用虚函数)1**

6. 阅读下面程序，写出输出结果。

```

#include <iostream>
using namespace std;

```

```

class Integer
{

```

```

private:
    int num;

public:
    Integer(int n = 0): num(n){ }
    void Set(int n) { num = n; }
    int Get() const { return num; }
    Integer operator++() { return num + 10 ; }
    Integer operator++(int) { return num + 18; }
};

int main()
{
    Integer i, j;

    i.Set(6);
    cout << i.Get() << endl;
    j = i; ++j;
    cout << j.Get() << endl;
    j = i++;
    cout << j.Get() << endl;
    j = ++i;
    cout << j.Get() << endl;
    cout << "end" << endl;

    return 0;
}

```

上面程序的输出结果为： 6 6 24 16

四、程序填空题（本大题共 3 小题，每小题 4 分，共 12 分）

3. 下列程序的输出结果为：

0

1

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 GetCount() { cout << count << endl; }
};

int Point::count = 0;                                // 静态数据成员的初始化

int main(void)
{
    Point::GetCount();
    Point a(6, 8);
    Point::GetCount();

    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:
    _____ Show() const { cout << 1 << endl; }
};

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

int main(void)
{
    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 Int
{
private:
    int n;

public:
    Int(int m): n(m) { }
    Int _____(const Int &a) { return Int(n * a.n); } // 重载乘法运算
    void Show() const { cout << n << endl; }
};

int main()
{
    Int a(3), b(6), c = a * b;
    c.Show();

    return 0;
}

```

符

}

五、编程题（本大题共 28 分）

设计一个虚基类 **Staff**（员工），包含编号、姓名和年龄保护数据成员以及相关的成员函数；由 **Staff** 派生出工程师类 **Engineer**，包含专业和职称保护数据成员以及相关的成员函数；再由 **Staff** 派生出领导类 **Leader**，包含职务和部门保护数据成员以及相关的成员函数；然后由 **Engineer** 和 **Leader** 类派生出主任工程师类 **Chairman**。

设计一个利用文件处理方式实现对公司人员（包括工程师、领导和主任工程师）进行管理，具有增加数据、更新数据、查询数据、删除数据以及重组文件的功能。

要求将公司人员信息存储在数据文件 **staff.dat** 中，为存储方便，定义索引文件，在索引文件中，每一项包括删除标志、编号、记录在数据文件 **staff.dat** 中的相对位置和人员类型，使用索引对数据文件进行定位操作。

试
卷 编
号：