

## 上海交通大学试卷(A)参考答案

(2009 至 2010 学年 第二学期)

一 班级号\_\_\_\_\_ 学号\_\_\_\_\_ 姓名 \_\_\_\_\_  
课程名称 C++程序设计 成绩 \_\_\_\_\_

- 1、类 CStudent 的拷贝构造函数的声明语句为 D。  
A. CStudent &CStudent (const CStudent other)  
B. CStudent CStudent(const CStudent other)  
C. CStudent (CStudent \*other)  
D. CStudent (const CStudent &other)
- 2、类的友元函数能访问该类的 C。  
A. 私有成员      B. 保护成员      C. 所有成员      D. 公有成员
- 3、下面关于静态数据成员的描述中，正确的是 A。  
A. 静态数据成员可以直接用类名调用  
B. 静态数据成员可以在类体内进行初始化  
C. 静态数据成员不能受 private 控制符的作用  
D. 类的不同对象有不同的静态数据成员值
- 4、当使用 fstream 流类定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为 C。  
A. ios::in      B. ios::out      C. fstream::in | fstream::out      D. 没有指定打开方式
- 5、若 char p[20]="hello world";则输出该字符串正确的语句是 D。  
A. cout<<p[20];      B. cout<<&p;      C. cout<<\*p;      D. cout<<p;
- 6、在派生类中重新定义虚函数时，除了 B 方面，其他方面都必须与基类中相应的虚函数保持一致。  
A. 参数个数和类型      B. 函数体      C. 函数名称      D. 返回类型
- 7、类模板定义如下：  
template <class T, int low, int high>  
class Array {...};  
对该类模板实例化正确的是 A。  
A. Array<float, 0, 20> x;      B. Array<int, int, int> x;  
C. template<int, 0, 20> x;      D. Array<int, 0, int> x;

我承诺，我将严格遵守考试纪律。

承诺人：\_\_\_\_\_

题号			
得分			
批阅人(流水阅卷教师签名处)			

8、公有成员提供了类对外部的接口，私有成员是类的内部实现，而 D 不许外界访问，但允许派生类的成员访问，这样既有一定的隐藏能力，也提供了开放的接口。

A. 公有成员      B. 私有成员      C. 私有成员函数      D. 保护成员

9、假定 AB 为一个类，则执行 AB a(2), b[3], \*p; 语句时共调用该类构造函数的次数为 C。

A. 1                      B. 3                      C. 4                      D. 5

10、如果 A 是已经定义好的一个类，函数 f 的原型为 A f()。r2 是 A 类的一个对象，在函数 f 中执行 return r2 时，系统将自动调用 B。

A. 缺省的构造函数      B. 拷贝构造函数      C. 赋值运算符重载函数      D. 不调用任何函数

## 二. 看程序，写结果（每题 5 分，共 35 分）

1、请写出下列程序运行结果

```
class ADD
```

```
{    friend ADD operator++(ADD op);
    friend ADD operator++(ADD &op, int n);
```

```
public:
```

```
    ADD(int i = 0, int j = 0) {a = i; b = j;}
    void Show() const {cout << "a=" << a << ",b=" << b << endl;}
```

```
private:
```

```
    int a, b;
```

```
};
```

```
ADD operator++(ADD op)
```

```
{    ++op.a;  ++op.b;  return op;}
```

```
ADD operator++(ADD &op, int n)
```

```
{    ++op.a;  ++op.b;  return op;}
```

```
void main()
```

```
{    ADD obj(1, 2);
    obj.Show();  (obj++).Show();  obj.Show();
    (++obj).Show();  obj.Show();
}
```

```
a=1, b=2
a=2, b=3
a=2, b=3
a=3, b=4
a=2, b=3
```

2、请写出下列程序运行结果

```

class CConAndDecon {
public:
    CConAndDecon(char value) {
        m_data = value;
        cout << "Object " << m_data << " constructor" << endl;
    }

    CConAndDecon(const CConAndDecon &other) {
        m_data = other.m_data - 1 ;
        cout << "Object " << m_data << " copy constructor" << endl;
    }

    CConAndDecon operator=(const CConAndDecon &right) {
        if( this != &right )
        { m_data = right.m_data + 1 ;
          cout << "Object 's new value is " << m_data << " " << endl;
        }
        return *this;
    }

    ~CConAndDecon()
    { cout << "Object " << m_data << " destructor" << endl; }
}

```

private:

```

    char m_data;
};

```

```

void Func(CConAndDecon x);

```

```

int main()
{
    CConAndDecon *p = new CConAndDecon('h');
    static CConAndDecon c1( 'k');
    Func(*p);    delete p;
    return 0;
}

```

```

void Func(CConAndDecon x)
{
    static CConAndDecon c1 = x;
    CConAndDecon c2 = c1;
}

```

3、请写出下列程序运行结果

Object h constructor
Object k constructor
Object g copy constructor
Object f copy constructor
Object e copy constructor
Object e destructor
Object g destructor
Object h destructor
Object f destructor
Object k destructor

```
class CMake
{
public:
    CMake(int n)
    {
        m_data = n ;
        cout << "构造 " << m_data << endl;
    }
    CMake(const CMake &obj)
    {
        m_data = obj.m_data + 1;
        cout << "拷贝构造" << m_data << endl;
    }
    ~CMahe() { cout << "析构 " << m_data << endl;}
    operator int() const { return m_data; }
private:
    int m_data;
};
```

```
CMahe MakeObject( int n )
```

```
{
    CMake  p (n);
    return  p;
}
```

```
int main()
```

```
{
    cout << MakeObject( 7 ) << endl;

    return 0;
}
```

构造 7
拷贝构造 8
析构 7
8
析构 8

4、请写出下列程序运行结果

```
class BaseFly
{ public:
    virtual void Fly() { cout << "\n---Class BaseFly::Fly()---\n"; }
};

class BirdFly: public BaseFly
{ public:
    void Fly() { cout << "----Class BirdFly::Fly()----\n" ;}
};

class DragonFly: public BaseFly
{ public:
    void Fly() { cout << "\n---Class DragonFlyFly::Fly()---\n"; }
};

void main()
{   BaseFly *pBase, oBase;
    BirdFly *pBird = new BirdFly();
    pBase = pBird;
    cout << "\nBirdFly->";
    pBase->Fly();
    DragonFly *pDragon = new DragonFly();
    pBase = pDragon;
    oBase = *pDragon;
    pBase->Fly();
    pBird->Fly();
    pDragon->Fly();
    oBase.Fly();
}
```

```
BirdFly->----Class BirdFly::Fly()----
----Class DragonFlyFly::Fly()----
----Class BirdFly::Fly()---

----Class DragonFlyFly::Fly()----

----Class BaseFly::Fly()---
```

5、写出下列程序执行结果

```
class Point
{
    friend bool operator!=(const Point &p1, const Point &p2)
    { return p1.x+p1.y != p2.x+p2.y ; }

private:
    int x, y;

public:
    Point(int a = 1, int b = 1)
    {
        x = a; y = b;
        cout << "构造 Point(" << x << ", " << y << ")" << endl;
    }
    Point(const Point &p)
    {
        x = p.x ; y = p.y ;
        cout << "拷贝构造 Point(" << x << ", " << y << ")" << endl;
    }
    ~Point() { cout << "析构 Point(" << x << ", " << y << ") " << endl; }
    Point &operator++()
    {
        if (x < y) ++x;
        else ++y;
        return *this;
    }
    void show() { cout << " Point(" << x << ", " << y << ")" << endl; }
    int getx() const {return x ;}
    int gety() const {return y ;}
};

int main( )
{
    const Point origin(10, 5);
    Point point2(6, 7);
    int n = 0;
    while (point2 != origin) {++point2; ++n;}
    point2.show();
    cout << "n= " << n << endl;
    return 0;
}
```

```
构造 Point (10,5)
构造 Point (6,7)
Point(7,8)
n= 2
析构 Point (7,8)
析构 Point (10,5)
```

6、请写出下列程序运行结果

```
void func( int );
```

```
int main()
```

```
{   for (int i = 30; i > 0; i /= 3)
```

```
    try { func(i);
```

```
        cout << "i = " << i << endl;
```

```
    } catch( int) { cout << "exception:  int " << endl;  }
```

```
        catch (double) { cout << "exception: double " << endl;  }
```

```
    return 0;
```

```
}
```

```
void func(int num )
```

```
{   if ( num % 3 ) throw 3;
```

```
    else if ( num % 5 ) throw 5.5;
```

```
}
```

i = 30

exception: int

exception: double

exception: int

7、写出下列程序的输出结果

```
template <class T>
```

```
class Sample
```

```
{protected:
```

```
    T  n;
```

```
public:
```

```
    Sample(T i) { n = i; cout << "construct " << n << endl;}
```

```
    ~Sample() { cout << "destruct " << n << endl;}
```

```
    void disp(){cout << "n=" << setfill ( '#' ) << setw ( 10 ) << n << endl;}
```

```
};
```

```
template <class T >
```

```
class model: public Sample<T>
```

```
{   T  m;
```

```
public:
```

```
    model(T  t1, T  t2): Sample<T>(t1) {m = t2; cout << "construct " << m<< endl; }
```

```
    ~model() { cout << "destruct " << m<< endl; }
```

```
    void disp(){cout << "n=" << setfill ( '#' ) << setw ( 10 ) << n << ' ' << m << endl;}
```

```
    operator T () const { return n + m;}
```

```
};
```

```
int main()
```

```
{   model<int> s (20, 30);
```

```
    s.disp();
```

```
    cout << (int) s << endl;
```

```
    return 0;
```

```
}
```

construct 20

construct 30

n=#####20 30

50

destruct 30

destruct 20

### 三. 程序填空（每空 2 分，共 30 分）

1、下列程序的输出是：

2

5 7

14

请填空。

```
class CConst {
public:
    CConst(int d = 0): len(d) { size += d; }
    void Print() const { cout << len << " " << size << endl; }
    static void show() { cout << size << endl; }
private:
    const int len;
    static int size;
};
```

```
int CConst::size = 2;
int main()
{
    CConst:: show();
    const CConst c(5) ;
    c.Print();
    CConst c2(7);
    c2.show();
    return 0;
}
```

2、下面是处理二维平面上线段的类，其中的 Point 是上一大题第 5 题中定义的 Point 类，请填空。

```
class line {
    Point start;
    Point end;
public:
    line(int sx, int sy, int ex, int ey): start(sx,sy), end(ex, ey) {}
    int length() // 计算线段的长度
    {
        return sqrt((end.getx()-start.getx())*(end.getx()-start.getx())+(end.gety()-start.gety())*(end.gety()-start.gety()));
    }
};
```



- 3、下面的函数打开一个保存着一批字符串的文本文件，字符串间用空格分离，各字符串的长度均小于 20。文件名作为参数传入函数。函数读出文件中的字符串，并输出字符串到屏幕，每行一个，最后统计输出在文件中一共有多少个字符串。

```
void FR(char* fname)
{
    ifstream fin(fname);
    char a[20];
    int cnt = 0;

    if (___!fin___) {
        cout << "can't open file";
        return;
    }
    while (___fin >> a___) {
        cout << a << endl;
        ___++cnt___;
    }
    ___fin.close()___ ;
    cout << "字符串总数为 " << cnt << endl;
}
```

- 4、补充函数 equal 使得程序结果为：

$5 + 6 = 11;$

$1.111 + 2.222 = 3.333$

```
template <class T>
T add (___T x, T y___)
{ ___return x + y___ ; }
```

```
int main()
{
    int i = 5, j = 6;
    double y1 = 1.111222, y2 = 2.222222222222;

    cout << i << " + " << j << " = " << add(i, j) << endl;
    cout << setprecision(4) << y1 << " + " << y2 << " = " << add(y1, y2) << endl;
    return 0;
}
```

#### 四. 编程（共 25 分）

- 1、设计一个学生类 `student`，包括姓名和三门课程成绩，利用重载运算符“+”将所有学生的成绩相加放在一个对象中，再对该对象求各门课程的平均分。即，运行下面测试程序时，能得到相应的执行结果。（10 分）

```
void main()
{
    student s1("Li", 78, 82, 86), s2("Zheng", 75, 62, 89);
    student s3("Ma", 89, 87, 95), s4("Xu", 54, 78, 66), s;
    cout << "输出结果" << endl;
    s1.disp(); s2.disp(); s3.disp(); s4.disp();
    s=s1 + s2 + s3 + s4; // 调用重载运算符
    avg(s, 4); // 友元函数求平均分
}
```

本测试程序的执行结果如下：

输出结果：

Li 78 82 86

Zheng 75 62 89

Ma 89 87 95

Xu 54 78 66

平均分 74 77 84

评分标准：

正确定义数据成员（2 分）：姓名和三门课的成绩

正确定义成员函数（3 分）：构造函数，`disp` 函数，`avg` 函数

正确定义+重载函数（1 分）：定义为成员函数或友元函数

每个函数的实现：各 1 分

2、编写一个程序，计算扇形面积和球体表面积。

已知：圆周率 = 3.1415926 且并定义为所有对象共享的常量

扇形面积 = 圆周率\*半径的平方\*角度/360

球体表面积 = 4\*圆周率\*球体半径的平方

要求：你需要从一个抽象类 container 出发，完成对扇形类（sector）和球体类（sphere）的设计。（10 分）

需通过如下的 main 函数：

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ container *bptr;    //定义抽象类指针 bptr
```

```
  sphere s_obj(4);    //创建球体对象 s_obj，半径为 4;
```

```
  sector c_obj(2,270); //创建圆柱体对象 c_obj，半径为 2，角度为 270 度
```

```
  bptr = &s_obj;
```

```
  cout << "球体表面积: " << bptr->area() << endl;
```

```
  bptr = &c_obj;
```

```
  cout << "扇形面积: " << bptr->area() << endl;
```

```
}
```

得到的屏幕输出应为：

球体表面积：201.062

扇形面积：9.423

评分标准：

正确定义基类（2 分）：一个全局共享的常量 PI 和纯虚函数 area

正确定义扇形类（2 分）

正确定义球类（2 分）

正确实现各成员函数：每个成员函数 1 分

3、整型数组 `int a[10]` 中随机地存放有数字 0~9（数字可以重复）。

现请设计一个类，它的功能是在数组 `a` 中顺序地抽取 5 个数字，使这 5 个数字组成的 5 位数为最大（注意：这 5 个数字的先后顺序必须同其在原数组 `a[10]` 中的先后顺序相同），返回这个五位数。（5 分）

类的定义如下：

```
class max {  
    int data[10];  
public:  
    max(int *a);  
    int result();  
};
```

如果数组 `a[10] = { 4, 7, 8, 0, 8, 6, 2, 4, 9, 1 }`，定义对象 `max m(a)`，并执行语句 `a.result()`，则返回值为 88691。请补充构造函数和 `result` 函数。

评分标准：构造函数的实现    1 分

`result` 函数的实现    4 分