# 计算机科学与技术学院 2016-2017 学年第 1 学期　考试试卷

## 面向对象程序设计　　A 卷　开卷

姓名_____班级_____学号_____考试日期_____

| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 总分 | 核对人 |
|---|---|---|---|---|---|---|---|---|
| 题分 | 15 | 20 | 20 | 15 | 15 | 15 | 100 | 马光志 |
| 得分 | | | | | | | | |

| 得分 | 评卷人 |
|---|---|
| | |

一、选择题(每小题 3 分，共 15 分)
答题卡见右，以答题卡为准。

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| C | B | B | A | D |

**1.** 对于如下程序：

```
#include <stdio.h>
struct A{
    virtual int f( ) { printf("A"); }
    virtual int g( ) { printf("B"); }
} a, *p;
struct B: A {
    int f( ) { A::f( ); printf("C"); }
    int g( ) { printf("D"); }
};
void main( ) { p=&b; p->f( ); p->g( ); }
```

则程序的输出为_____：

**A. AB**　　**B. CD**

**C. ACD**　　**D. ABCD**

**2.** 如下程序：

```
#include <stdlib.h>
struct A { A( ) { } }a;
void main( ) { A b; exit(0 ); }
```

则关于对象 a, b 的析构，如下哪个叙述正确_____：

**A.** 无析构函数都没有析构;　　**B.** a 析构了但是 b 没析构;

**C.** b 析构了但是 a 没析构;　　**D.** 都析构了;

**3.** 对于定义“const char *&g( );”，如下哪个语句是错误的_____:

**A.** g( )= "abcde";　　**B.** *g(　)= 'A';

**C.** const char *p=g(　);　　**D.** const char *&q =g(　);

**4.** 对于如下定义：

```
struct A { virtual int f( ) { return 1;} }a;
struct B: A {
    int f( ) const { return 2; }
    int f( ) volatile { return 3; }
    int f( ) const volatile { return 4; }
} c;
int main(int argc, char* argv[ ]) { A *p = &c;   return p->f( ); }
```

主函数 main 的返回值是_____:

A. 1　　B. 2

C. 3　　D. 4

5. 关于运算符重载的叙述哪个正确_____:

A. 可以改变优先级和结合性　　B. 可以改变优先级，不能改变结合性

C. 不能改变优先级，可改变结合性 D. 都不能改变

| 得分 | 评卷人 |
|---|---|
| | |

二、在作用域运算最多允许一层（即不允许两层运算如 A::B::c 及两层以上运算）的情况下，指出各类可访问的成员及其访问权限(20)。

```
class A{                    |A 的成员
    int   a;                |private:
protected:                  |     int a;
    int   b, e;             |protected:
public:                     |     int b, e;
    int   c, d;             |public:
};                          |     int c, d;

class B: protected A{       |B 的成员
    int   d;                |private:
protected:                  |     int d;
    int   c, e;             |protected:
public:                     |     int c, e, A::(b, d, e)
    A::c;                   |public:
    int   f, b;             |     int f, b, A::c
};

class C: public A {         |C 的成员
    int   g;                |private:
protected:                  |     int g;
```

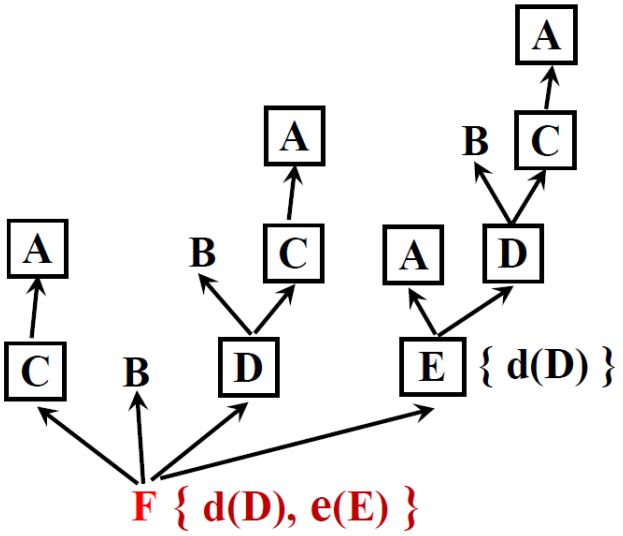
```
    int   h, d;              |protected:
public:                      |     int h, d, A::(b, e)
    int   i;                 |public:
};                           |     int i, A::(c, d)
struct D: protected B, C{    |D 的成员
    int   j;                 |private:
protected:                   |protected:
    int   k;                 | int k,B::(c,e,f,b,A::(b,c,d,c)), C:(h,d, A::(b,c))
public:                      |public:
    int   n, d;              |     int j, n, d, C::(i, A:(c, d))
};
```

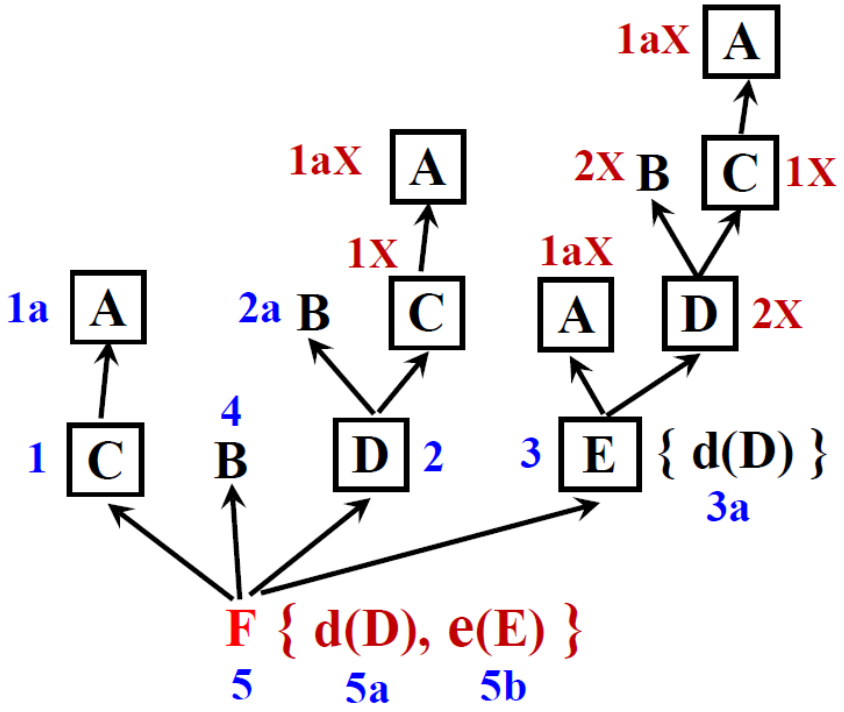
| 得分 | 评卷人 |
| --- | --- |
| | |

三、指出 main 中每行的输出结果 (前四题每题 3 分，后两题每题 4 分，共 20 分)

```
#include <iostream.h>
struct A {A( ) { cout<<'A'; } };
struct B {B( ) { cout<<'B'; } };
struct C: virtual A { C( ) { cout<<'C'; } };
struct D: B, virtual C { D( ) { cout<<'D'; } };
struct E: virtual A, virtual D {
    D   d;
    E( ): A( ) { cout<<'E'; }
};
struct F: virtual C, B, virtual D, virtual E {
    D d;   E e;
    F( ) { cout<<'F'; }
};
void main( ) {
    A   a;   cout<<'\n';   //输出=A
    B   b;   cout<<'\n';   //输出=B
    C   c;   cout<<'\n';   //输出=AC
    D   d;   cout<<'\n';   //输出=ACBD
    E   e;   cout<<'\n';   //输出=ACBDACBDE
    F   f;   cout<<'\n';   //输出=ACBDACBDEBACBDACBDACBDEF
}
```

下面给出 **F f** 的分析：




F 输出：ACBDACBDEBACBDACBDACBDEF

E 输出：ACBD ACBD E

同名的虚基类只能构造 1 次；

从最左边的子树开始扫描虚基类，并进行构造。

| 得分 | 评卷人 |
| --- | --- |
| | |

四、指出以下程序的语法错误及其原因 (每错约 1 分，共 15 分)

```
class A {
    int &a;
protected:
    const int &b;
    ~A( ) { }
public:
    int c;
    virtual A&   (*g)( );  //(1)virtual 不能用于定义数据成员
    A(int x) { b = x; };   //(2)没在函数体前初始化 a 和 b, (3)不能在体内初始 b
} a = (1, 2, 3);           //(4)a 不能调用私有的析构函数~A( )

class B: A {
    int d;
public:
    A::b;                  //(5)不能改变访问权限，只能恢复（有问题）
    static int operator( )(int) { return 3; }; //(6)运算符( )只能为实例函数成员
    B(int x, int y, int z):A(x) { d=x+y+z; };
} b(2, 3, 7);

struct C: B{               //(7)编译程序无法自动生成构造函数 C( )
    int z;
protected:
    virtual ~C( ) {   };
} c;                       //(8)c 不能调用保护的析构函数~C( )

void   main( ) {
    int *A::*p = &c.z;     //(9)不能将 int *类型的值赋给 int *A::*类型的变量 p
    int   i = a.b;         //(10)main 不能访问保护成员 A::b（有问题）
    i = a;                 //(11)对象 a 无法转换为整数
    i = b.b;               //(12) main 不能访问私有成员 B::b
    i = c.d;               //(13) main 不能访问私有成员 C::d
    i = b.*p;              //(14)不能将 int *类型的值赋值给整型变量 i
    return 1;              //(15)main 无需返回值
}
```

| 得分 | 评卷人 |
| --- | --- |
| | |

五、请填入学号最后一位十进制数字，指出 main 函数中变量 i 在每条赋值语句执行后的值 (每小题 2.5 分，共 15 分)

```
int    x = 学号最后一位十进制数      ,  y=x+3;
struct A {
    int x;
    static int &y;
public:
    operator int( )const { return x + y; }
    int &v(int &x) {
        for(int y=1; x<201; x^=y, y++)
            if(x>200) { x -= 31; y -= 2;}
         return ++x;
    }
    A &operator++( ) { ++x; ++y; return *this; }
    A(int x = ::x + 2, int y = ::y + 3) { A::x = x;   A::y = y; }
};

int & A::y = ::x;

void main( ) {
    A   a(2, 3),   b(a),   c;
    int   i,   &j=i,   A::*p=&A::x;
    i = a.y;                    //i=
    j = a.x++;                  //i=
    i = a.*p;                   //i=
    i = ++a;                    //i=
    i = b.y + ::y;              //i=
    (b.v(i) = 2) += 3;          //i=
}
```

答：

| 学号 | i=a.y | j=a.x++ | i=a.*p | i=++a | i=b.y+::y | (b.v(i)=2)+=3 |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 6 | 2 | 3 | 11 | 10 | 5 |
| 1 | 7 | 2 | 3 | 12 | 12 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 8 | 2 | 3 | 13 | 14 | 5 |
| 3 | 9 | 2 | 3 | 14 | 16 | 5 |
| 4 | 10 | 2 | 3 | 15 | 18 | 5 |
| 5 | 11 | 2 | 3 | 16 | 20 | 5 |
| 6 | 12 | 2 | 3 | 17 | 22 | 5 |
| 7 | 13 | 2 | 3 | 18 | 24 | 5 |
| 8 | 14 | 2 | 3 | 19 | 26 | 5 |
| 9 | 15 | 2 | 3 | 20 | 28 | 5 |

**分析：**

| | ::y | ::x, A::y | a.x | b.x | c.x | i |
|---|---|---|---|---|---|---|
| | n+3 | n | | | | |
| a(2, 3) | | 3 | 2 | | | |
| b(a) | | 3 | | 2 | | |
| c <=> A(5, n+6) | | n+6 | | | 5 | |
| i = a.y | | | | | | n+6 |
| j = a.x++ | | | 3 | | | 2 |
| i = a.*p | | | | | | 3 |
| i = ++a | | n+7 | 4 | | | n+11 |
| i = b.y + ::y | | | | | | 2n+10 |
| (b.v(i)=2) += 3 | | | | | | 5 |

| 得分 | 评卷人 |
|---|---|
| | |

六、N 个顶点的无向图 MAP 最多有 N*(N-1)条边，设顶点的编号为 0, 1, …, N-1，每条边由其中任意两个顶点连接而成，试定义如下无向图类中的成员函数。(每小题 2.5 分，共 15 分)

```
class MAP {
    int (*const e)[2];    //边集指针 e,边 x 的顶点为 e[x][0]和 e[x][1]
    const int n;          //图的顶点个数
```

```
    int c;                  //图实际已有的边的个数
public:
    MAP(int n);    //图最多 n 个顶点，假设图初始时无边
    MAP(const MAP& m);                  //深拷贝构造函数
    MAP &operator=(const MAP& m);   //深拷贝赋值函数
    MAP &operator( )(int v0, int v1);   //连接顶点 v0 和 v1 成边,设 v0<v1
    int (*operator[ ](int x))[2];       //取图中的边 x
    ~MAP( );                            //析构函数
};
```

**答：**

```
MAP::MAP(int n): e(new int[n*(n-1)][2]),n(e? n:0), c(0) { }

MAP::MAP(const MAP& m): e(new int[m.n*(m.n-1)][2]),n(e?m.n:0){
    for(c=0; c<m.c; c++){
        e[c][0] = m.e[c][0];
        e[c][1] = m.e[c][1];
    }
}

MAP& MAP::operator=(const MAP &m) {
    if(e) delete e;
    *(int (**)[2])&e = new int[m.n*(m.n-1)][2];
    *(int *)&n = e?m.n:0;
    for(c=0; c<m.c; c++){
        e[c][0] = m.e[c][0];
        e[c][1] = m.e[c][1];
    }
    return *this;
}

MAP& MAP::operator( )(int v0, int v1) {//连接顶点 v1 和 v2 成为边
    int t=v0;
    if(v0==v1 || v0<0 || v0>=n || v1<0 || v1>=n) return *this;
```

```
    if(v0>v1) { v0 = v1; v1 = t; }
    for(t=0; t<c; t++)    if(e[t][0]==v0 && e[t][1]==v1) return *this;
    e[c][0] = v0;
    e[c++][1] = v1;
    return *this;
}

MAP::~MAP( ) {    //析构函数
    if(e) { delete e; *(int (**)[2])&e=0; *(int *)&n=0; c=0; }
}

int (*MAP::operator[ ](int x)) [2] {   return e + x;   }
```