

## 面向对象程序设计模拟试卷一

### 一. 单选题(15)。

- 关于构造的叙述\_\_\_\_\_正确:
  - 最先构造虚基类
  - 最先构造基类
  - 最先构造派生类的对象成员
  - 都不对
- 关于静态数据成员的叙述\_\_\_\_\_正确:
  - 公有的可在类体外初始化
  - 私有的不能在类体外初始化
  - 私有和保护的不可以在类体外初始化
  - 都可以且必须在体外初始化
- 若派生类函数不是基类的友元, 关于该函数访问基类成员\_\_\_\_\_正确:
  - 公有的可被派生类函数访问
  - 都可以被派生类函数访问
  - 公有和保护的可被派生类函数访问
  - 都不对
- 关于函数的所有缺省参数的叙述\_\_\_\_\_正确:
  - 只能出现在参数表的最左边
  - 只能出现在参数表的最右边
  - 必须用非缺省的参数隔开
  - 都不对
- 使用 friend、virtual、static 说明函数的叙述\_\_\_\_\_正确:
  - 必须同时使用三个
  - 只能同时用其中两个
  - 只能独立单个地使用
  - 都不对

### 二. 指出各类的成员及其存取属性(20) 。

```
class A{
    int a;
protected:
    int b;
public:
    int c;
};
class B: protected A{
    int d;
protected:
    int e;
public:
    A::c;
    int f;
};

class C: A{
    int g;
protected:
    int h;
public:
    int i;
};
class D: B, C{
    int j;
protected:
    B::b;
    int k;
private:
    int n;
};
```

### 三. 指出 main 中每行的输出结果(20) 。

```
#include <iostream.h>
struct A{A(){ cout<<'A';}};
struct B{B(){ cout<<'B';}};
struct C: A{C(){ cout<<'C';}};
struct D: virtual B, C{D(){ cout<<'D';}};
struct E: A{
    C c;
    E(): c(){ cout<<'E';}
};
```

```

struct F: virtual B, C, D, E{
    F(){ cout<<'F';}
};
void main(){
    A  a;  cout<<"\n";
    B  b;  cout<<"\n";
    C  c;  cout<<"\n";
    D  d;  cout<<"\n";
    E  e;  cout<<"\n";
    F  f;  cout<<"\n";
}

```

四. 指出以下程序的语法错误及其原因(15) 。

```

class A{
    static int  a=0;
protected:
    int  b;
public:
    int  c;
    A(int);
    operator int( );
} a(1, 2);
class B: A{
    B(int);
    virtual int d;
    int  e;
public:
    A::b;
    friend int operator =(B);
    static B(int, int);
} b=5;
class C: B{
public:
    int operator++(double);
};
int main( ){
    int  *A::*p, i;
    i=a.a;
    i=A(4);
    i=b.c;
    p=&A::c;
    i=b;
    return ;
}

```

五. 指出 main 变量 i 在每条赋值语句执行后的值(15) 。

```

int    x=2,  y=x+30;
struct A{
    static int  x;
    int  y;
public:
    operator int( ){ return x-y; }
    A operator ++(int){ return A(x++, y++); }
    A(int x::x+2, int y::y+3){ A::x=x;  A::y=y; }
    int &h(int &x);
};
int &A::h(int &x)
{
    for(int y=1; y!=1|| x<201; x+=11, y++)  if(x>200) { x-=21; y-=2;}
    return x-=10;
}
int A::x=23;
void main(){
    A  a(54, 3),  b(65),  c;
    int  i,  &z=i,  A::*p=&A::y;
    z=b.x;
    i=a.x;
    i=c.*p;
    i=a++;
    i=:x+c.y;
    i=a+b;
    b.h(i)=7;
}

```

六. 为了没有误差地表示分数，定义类 **FRACTION** 来表示分数，请编程实现分数类中的除 **cmd** 外的所有函数(15)。

```

class FRACTION{ //对于  $\frac{6}{7}$ ，numerator 存分子 6，denominator 存分母 7
    int    numerator, denominator;
    static int cmd(int x, int y);           //求整数 x,y 的最大公约数
public:
    int operator>(const FRACTION&)const;    //大于比较，例  $\frac{6}{7} > \frac{2}{3}$ 
    FRACTION(int num, int den=1);           //num、den 各为分子和分母
    FRACTION operator*( )const;             //分数约简， $*\frac{30}{36} = \frac{5}{6}$ 
    FRACTION operator+(const FRACTION&)const; //加法， $\frac{6}{7} + \frac{2}{3} = \frac{32}{21}$ 
    FRACTION operator*(const FRACTION&)const; //乘法， $\frac{6}{7} * \frac{2}{3} = \frac{12}{21} = \frac{4}{7}$ 
};
int  FRACTION::cmd(int x, int y){
    int r;
    if(x<y){  r=x;  x=y;  y=r;  }
    while(y!=0){  y=x%(r=y);  x=r;  }
}

```

```
return x;
```

```
}
```

## 面向对象程序设计模拟试卷一答案

一解： 1. A    2. D    3. C    4. B    5. C

二解： 各类的成员及其存取属性如下。

1.private:	2.private:
int a;	int d;
protected:	protected:
int b;	int e;
public:	int b;
int c;	public:
	int c;
	int f;
3.private:	4.private:
int g;	int n, j;
int b;	int e, c;
int c;	int f, h;
protected:	int i;
int h;	protected:
public:	int b;
int i;	int k;

三解： main 中每行的输出结果见如下注释。

```
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 BACD
E e; cout<<"\n";    //输出 AACE
F f; cout<<"\n";    //输出 BACACDAACEF
```

四解： 错误及其原因见注释。

```
class A{
    static int a=0;    //不能在类的体内初始化
protected:
    int b;
public:
    int c;
    A(int);
    operator int( );
} a(1, 2);    //没有定义A(int, int)
class B: A{
    B(int);
    virtual int d;    //virtual不能用于数据成员
    int e;
public:
    A::b;    //修改而不是恢复访问权限
```

```

    friend int operator =(B); //不存在普通赋值函数的重载，=必须重载为双目的
    static B(int, int);      //构造函数不能定义为不带this参数的静态函数成员
} b=5;                      //B(int)是不可访问的私有构造函数

class C: B{
public:
    int operator++(double); //必须用int定义后置运算
}; //C 必须定义构造函数，因 B 定义有带参构造函数且没有无参构造函数

int main( ){
    int *A::*p, i;
    i=a.a;                //a.a 是私有的，不能被不是 A 的友元的 main 访问
    i=A(4);
    i=b.c;                //b.c 是私有的，不能被不是 B 的友元的 main 访问
    p=&A::c;              //A::c 不是一个 int *指针
    i=b;                  //B 不存在公有的 B::operator int()或 B::operator A()
    return ;              //返回 void 类型，与 main 声明的返回类型不一致
}

```

五解：变量 i 在每条赋值语句执行后的值见注解。

```

z=b.x;        //i=4
i=a.x;        //i=4
i=c.*p;       //i=35
i=a++;        //i=1
i::x+c.y;     //i=37
i=a+b;        //i=-31
//A(4,3)++将 A::x=a.x=b.x=c.x 置为 4，普通成员 a.y=4 不变
b.h(i)=7;     //i=7

```

六解：本题所涉及的概念并不复杂，值得注意的是分数的约分运算，该运算实际上是一个单目运算。在重载约分运算时，要选择只有一个操作数的运算符。另外，在对分数进行加和乘以后，最好对运算结果进行约分运算，如果不约分不算错。函数成员如下。

```

FRACTION::FRACTION(int num, int den){
    numerator=num;
    denominator=den;
}

int FRACTION::operator>(const FRACTION&f)const{
    int d= denominator *f.denominator;
    return (numerator*f.denominator/d >denominator*f.numerator/d);
}

FRACTION FRACTION::operator*( )const{
    int c=cmd(numerator, denominator);
    return FRACTION(numerator/c, denominator/c);
}

FRACTION FRACTION::operator+(const FRACTION&f)const{
    int n= numerator*f.denominator+denominator*f.numerator;
    int d= denominator*f.denominator;
    return *FRACTION(n, d); //对运算结果进行约分运算
}

```

```
}  
FRACTION FRACTION::operator*(const FRACTION&f)const{  
    return *FRACTION(numerator*f.numerator, denominator*f.denominator); //约分  
}
```