一、选择题

C B A A B

二、可访问成员和访问权限

A的可访问成员和访问权限

**private:**a

**protected:**b,f

**public:**c,d

B的可访问成员和访问权限

**private:**d,

**protected:**c,e,A::(b,f,c,d)

**public:**f

C的可访问成员和访问权限

**private:**g

**protected:**h,d

**public:**c,i,A::(b,f,c,d)

D的可访问成员和访问权限

**protected:**k,c

**public:**j,n,B::(c,e,f),A: :(b,f,c,d),C::(h,d,c,i)

三、

A a; A

B b; B

C c; AC

D d; ACBD

E e; ACBDAACBDE

F f; ACBDBAACBDEACBDF

四、改错题

(1) ~A( ) { };析构函数必须为pulic

(2) virtual A (\*g)(int);virtual不能定义数据成员

(3) A(int x) { a = x; };没有在成员列表中初始化成员b

(4)运算符()只能为实例函数成员

(5) ~C(int x) { z = x; };析构函数无参数

(6) int A::\*p = &c.z; C和A不构成父子关系，必须要进行强制类型转换

(7)x.b无法访问

(8)A无法转换为int类型

(9)C无法自动生成构造函数

(10) A::b;无法改变访问权限protected为public

(11)b.b无法访问

(12)c.d无法访问

五、执行后的值

x=3

4

11

4

17

24

5

六、写出成员函数

PER::PER(int p):n(p),f() //用自然数 p 初始化 n, f, c

{

int \*temp=new int[p],num=0,sum=0;

for (int i = 1; i < p; i++)

{

if (p % i == 0)

temp[num++] = i;

}

\*(const\_cast<int\*\*>(&f))= new int[num];

for (int i = 0; i < num; i++)

{

f[i] = temp[i];

sum += f[i];

}

if (sum == n)c = num;

else c = -1 \* num;

}

PER::PER(const PER& p):n(p.n),f() //深拷贝构造函数

{

\*(const\_cast<int\*\*>(&f)) = new int[p.n];

for (int i = 0; i < p.n; i++)

f[i] = p.f[i];

c = p.c;

}

PER& PER::operator=(const PER& p) //深拷贝赋值运算

{

if (f)

delete[]f;

\*(const\_cast<int\*\*>(&f)) = new int[p.n];

for (int i = 0; i < p.n; i++)

f[i] = p.f[i];

c = p.c;

return \*this;

}

PER::operator int() const //若不是完全数则返回 0，否则返回 c

{

if (c < 0)return 0;

else return c;

}

int PER::operator [ ] (int k) const //返回 k 所指示的因子, 若 k < 0 或 k >= c 返回 0

{

if (k < 0 || k >= c)return 0;

else return f[k];

}

PER::~PER() //析构函数

{

delete[]f;

}