

Exercises about Classes

Prof. Sai-Keung Wong

TA: xyz

Questions

```
class A
{
    public:
        A( );
    private:
        int x;
    protected:
        double y;
    public:
        string z;
};
```

```
void main( ) {
    A p;
    p.x = 10;
    p.y = 12.5;
    p.z = "here";
}
```

Which line(s) has(have)
an error?

Why?

Answers

```
class A
{
public:
    A( );
private:
    int x;
protected:
    double y;
public:
    string z;
};
```

```
void main( ) {
    A p;
    p.x = 10;
    p.y = 12.5; ✗
    p.z = "here"; ✗
}
```

Which line(s) has(have)
an error?

Why?

Questions

```
class A
{
public:
    A( );
private:
    int x;
protected:
    double y;
public:
    string z;
};
```

```
void main( ) {
    A p;
    A q, *x;
}
```

How many objects of A
are created?

Why?

Answers

```
class A
{
public:
    A( );
private:
    int x;
protected:
    double y;
public:
    string z;
};
```

```
void main( ) {
    A p;
    A q, *x;
}
```

How many objects of A
are created?

Two.

x is a pointer. p and
q are objects.

Questions

```
class A
{
    public:
        A( );
    private:
        int x;
    protected:
        double y;
    public:
        string z;
};
```

```
void main( ) {
    A p;
    A q, *x;
    new A;
    x = new A;
}
```

How many objects of A
are created?

Answers

```
class A
{
    public:
        A( );
    private:
        int x;
    protected:
        double y;
    public:
        string z;
};
```

```
void main( ) {
    A p;
    A q, *x;
    new A;
    x = new A;
}
```

How many objects of A
are created?

Four.

Questions

```
class A
{
public:
    A( ) { cout << "AC" << endl; }
private:
    int x;
protected:
    double y;
public:
    string z;
};
```

```
void main( ) {
    A p, *x = new A;
}
```

Any error?

If no error, what are the output?

Answers

```
class A
{
public:
    A( ) { cout << "AC" << endl; }
private:
    int x;
protected:
    double y;
public:
    string z;
};
```

```
void main( ) {
    A p, *x = new A;
    //instantiate
}
```

Any error? No.

If no error, what are the output?

AC

AC

Questions

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};
```

```
void main( ) {  
    A p;  
    B q;  
}
```

what are the output?

Answers

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};
```

```
void main( ) {  
    A p;  
    B q;  
}
```

what are the output?

AC

AC
BC }

Questions

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};  
class C : public A {  
public:  
    B( ) { cout << "CC" << endl; }  
};
```

```
void main( ) {  
    A p;  
    B q;  
    C w;  
}
```

Any error?

If there is an error, what is the error?

Answers

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};  
class C : public A {  
public:  
    C( ) { cout << "CC" << endl; }  
};
```

```
void main( ) {  
    A p;  
    B q;  
    C w;  
}
```

Any error?

Yes

The constructor of C.

Questions

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};  
class C : public A {  
public:  
    C( ) { cout << "CC" << endl; }  
};
```

```
void main( ) {  
    A p;  
    B q;  
    C w;  
}
```

Any errors?

If no, what are the output?

Answers

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};  
class C : public A {  
public:  
    C( ) { cout << "CC" << endl; }  
};
```

```
void main( ) {  
    A p;  
    B q;  
    C w;  
}  
what are the output?  
AC  
AC } for q  
BC  
AC } for w  
CC
```

Questions

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};  
class C : public B {  
public:  
    C( ) { cout << "CC" << endl; }  
};
```

```
void main( ) {  
    C w;  
}  
what are the output?
```


Answers

```
class A {  
public:  
    A( ) { cout << "AC" << endl; }  
};  
class B : public A {  
public:  
    B( ) { cout << "BC" << endl; }  
};  
class C : public B {  
public:  
    C( ) { cout << "CC" << endl; }  
};
```

```
void main( ) {  
    C w;  
}
```

what are the output?

AC

BC

CC

Questions

```
class A { public:  
    A( ) { cout << "AC" << endl; }  
    ~A( ) { cout << "DA" << endl; }  
};  
class B : public A { public:  
    B( ) { cout << "BC" << endl; }  
    ~B( ) { cout << "DB" << endl; }  
};  
class C : public B { public:  
    C( ) { cout << "CC" << endl; }  
    ~C( ) { cout << "DC" << endl; }  
};
```

```
void main( ) {  
    C *w = new C;  
    delete w;  
}  
what are the output?
```

Answers

```
class A { public:  
    A( ) { cout << "AC" << endl; }  
    ~A( ) { cout << "DA" << endl; }  
};  
class B : public A { public:  
    B( ) { cout << "BC" << endl; }  
    ~B( ) { cout << "DB" << endl; }  
};  
class C : public B { public:  
    C( ) { cout << "CC" << endl; }  
    ~C( ) { cout << "DC" << endl; }  
};
```

```
void main( ) {  
    C *w = new C;  
    delete w;  
}  
what are the output?  
AC  
BC  
CC  
DC  
DB  
DA
```

Questions

```
class A { public:  
    A( ) { cout << "AC" << endl; }  
    A( int a ) {  
        cout << "AC" << a << endl; }  
};  
  
class B : public A { public:  
    B( ) { cout << "BC" << endl; }  
    B( int b ) {  
        cout << "BC" << b << endl; }  
};
```

```
void main( ) {  
    A x, y(10);  
    B w(12);  
}  
what are the output?
```

Answers

```
class A { public:
    A( ) { cout << "AC" << endl; }
    A( int a ) {
        cout << "AC" << a << endl; }
};

class B : public A { public:
    B( ) { cout << "BC" << endl; }
    B( int b ) {
        cout << "BC" << b << endl; }
};
```

```
void main( ) {
    A x, y(10);
    B w(12);
}
```

what are the output?

AC

AC10

AC <- default constructor of A

BC12

Questions

```
class A { public:
    A( ) { cout << "AC" << endl; }
    A( int a ) {
        cout << "AC" << a << endl; }
};

class B : public A { public:
    B( ) { cout << "BC" << endl; }
    B( int b ) : A(), A( b ) {
        cout << "BC" << b << endl; }
};
```

```
void main( ) {
    A x;
    B y, w(12);
}
```

Any error?

If there is an error, what is the error?

Answers

```
class A { public:
    A( ) { cout << "AC" << endl; }
    A( int a ) {
        cout << "AC" << a << endl; }
};

class B : public A { public:
    B( ) { cout << "BC" << endl; }
    B( int b ) : A(), A( b ) {
        cout << "BC" << b << endl; }
};
```

```
void main( ) {
    A x;
    B y, w(12);
}
```

Any error?

Yes, there is an error.

Base class A has been initialized.

Questions

```
class A { public:  
    A( ) { cout << "AC" << endl; }  
    A( int a ) {  
        cout << "AC" << a << endl; }  
};  
class B : public A { public:  
    B( ) { cout << "BC" << endl; }  
    B( int b ) : A( ) {  
        cout << "BC" << b << endl; }  
};
```

```
void main( ) {  
    A x;  
    B y, w(12);  
}  
what are the output?
```


Answers

```
class A { public:
    A( ) { cout << "AC" << endl; }
    A( int a ) {
        cout << "AC" << a << endl; }
};

class B : public A { public:
    B( ) { cout << "BC" << endl; }
    B( int b ) : A( ) {
        cout << "BC" << b << endl; }
};
```

```
void main( ) {
    A x;
    B y, w(12);
}

what are the output?

AC
AC
BC
AC
BC12
```

Questions

```
class A { public:
    A( ) { cout << "AC" << endl; }
    A( int a ) {
        cout << "AC" << a << endl; }
};

class B : public A { public:
    B( ) { c = 20; cout << "BC" << endl; }
    B( int b ) : A( b ), c(18) {
        cout << "BC" << b << endl;
        cout << c << endl;
    }
    int c;
};
```

```
void main( ) {
    B y, w(12);
}
```

Any errors?

If there are any errors, explain clearly.

If there is no error, what are the output?

Answers

```
class A { public:
    A( ) { cout << "AC" << endl; }
    A( int a ) {
        cout << "AC" << a << endl; }
};

class B : public A { public:
    B( ) { c = 20; cout << "BC" << endl; }
    B( int b ) : A( b ), c(18) {
        cout << "BC" << b << endl;
        cout << c << endl;
    }
    int c;
};
```

```
void main( ) {
    B y, w(12);
}
```

```
AC
BC
AC12
BC12
18
```

Concept: Data member initialization

```
class A {  
    A( ) : x(12), y( "here"), ... {  
    }  
    int x;  
    string y;  
};
```

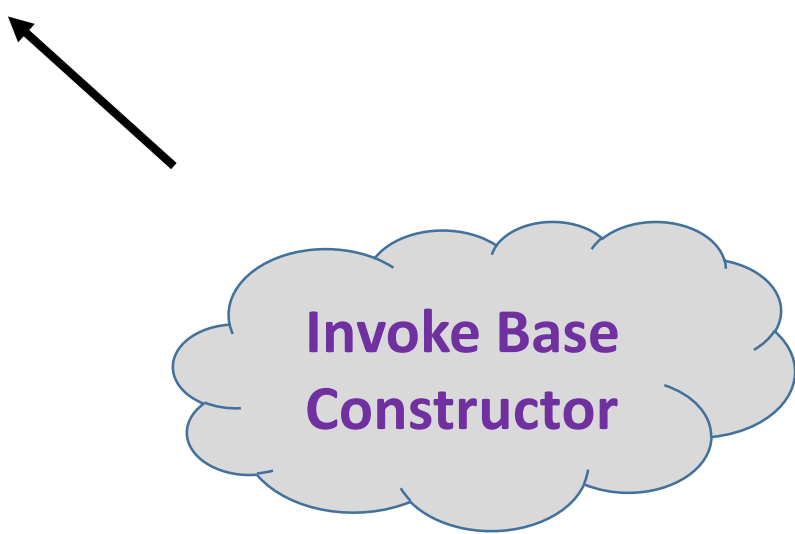
Concept: Data member initialization

```
class A {  
    A( ) : variable(value), variable(value), ....  
    {  
  
    }  
    int x;  
    string y;  
};
```

```
class A {  
    A( ) : x(12), y( "here"), ...  
    {  
  
    }  
    int x;  
    string y;  
};
```

Concept: Data member initialization

```
class B : public A, C {  
    B( ) : variable(value), variable(value), ...,  
          A( arg-list )  
    {  
  
    }  
    int x;  
    string y;  
};
```



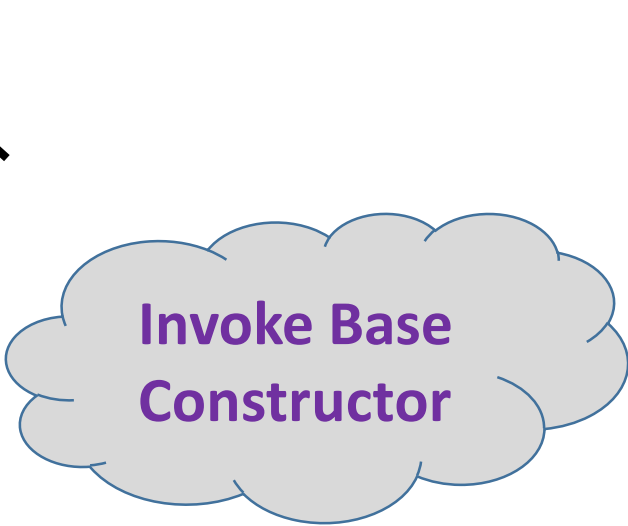
The diagram illustrates the concept of data member initialization. A cloud labeled "Invoke Base Constructor" has an arrow pointing to the base constructor call `A(arg-list)` in the code snippet.

```
class B : public A {  
    B( ) : x(12), y( "here" ), ...,  
          A( 10 )  
    {  
  
    }  
    int x;  
    string y;  
};
```

Concept

Data member initialization: **Can we invoke the base constructor?**

```
class B: public A {  
    B( ) : variable(value), variable(value), ...,  
          A( arg-list )  
    {  
  
    }  
    int x;  
    string y;  
};
```



The diagram illustrates the concept of invoking the base constructor. A cloud labeled "Invoke Base Constructor" has an arrow pointing to the "A(arg-list)" call in the constructor initialization list of class B.

```
class B : public A {  
    B( ) : x(12), y( "here" ), ...,  
          A( 10 )  
    {  
  
    }  
    int x;  
    string y;  
};
```

Questions

```
class A {  
    private:  
        static int x;  
    public:  
        A() { }  
};
```

- 1) How to initialize x? Explain clearly.
- 2) Implement a member function to increase x by 1.

Answers

```
class A {  
    private:  
        static int x;  
    public:  
        A() { }  
        void foo( ) { ++x; }  
};  
int A::x =0;
```

- 1) How to initialize x? Explain clearly.
- 2) Implement a member function to increase x by 1.

Answers

```
class A {  
private:  
    static int x;  
public:  
    A() { }  
    static void foo( ) { ++x; }    // static is also fine.  
};  
int A::x =0;
```

- 1) How to initialize x? Explain clearly.
- 2) Implement a member function to increase x by 1.

Questions

```
class A {  
    private:  
        static int x;  
    public:  
        A() { }  
        static void foo( ) { ++x; }  
};  
int A::x =0;  
How to call foo() ?
```

Answers

```
class A {  
    private:  
        static int x;  
    public:  
        A() { }  
        static void foo( ) { ++x; }  
};  
int A::x =0;  
How to call foo() ?
```

```
A x;  
x.foo();
```

or

```
A::foo();
```

Questions

```
class A {  
private:  
    static int x;  
    int y;  
public:  
    A() { y = 1 };  
    static void foo( ) { y = 0; ++x; }  
};  
int A::x =0;
```

What are the errors?

Answers

```
class A {  
    private:  
        static int x;  
        int y;  
    public:  
        A() { y = 1; }  
        static void foo( ) { y = 0; ++x; }  
};  
int A::x = 0;  
What are the errors?
```

foo() is a static method.

Non-static data members cannot be modified in a static method.

// ; must be put at the end of a statement

Questions

```
class A {  
private:  
    int *x;  
public:  
    A( ) { x = new int[1000]; }  
};
```

1) Implement the destructor of A to release the memory space pointed by x.

Answers

```
class A {  
private:  
    int *x;  
public:  
    A( ) { x = new int[1000]; }  
    ~A() { delete x; } // error. x is a pointer which points to an array  
};
```

1) Implement the destructor of A to release the memory space pointed by x.

Answers

```
class A {  
private:  
    int *x;  
public:  
    A( ) { x = new int[1000]; }  
    ~A() { delete [] x; }  
};
```

1) Implement the destructor of A to release the memory space pointed by x.

Concept: Singleton

```
class A {  
private: static A *_instance;  
public:  
    static A *instance( ) {  
        if ( _instance == 0 ) _instance = new A;  
        return _instance;  
    }  
};  
A *A::_instance = 0;
```

Concept: Singleton

```
class A {  
private: static A *_instance;  
public:  
    static float computeValue( ) { .....}  
    static A *_instance( ) {  
        if ( _instance == 0 ) _instance = new A;  
        return _instance ;  
    }  
};  
A *  
A::_instance = 0;
```

How to use it?

A::_instance()->functionName(...);

float x = A::_instance()->computeValue();

Concept

```
class A {  
    private:  
        static int x;  
        static int y;  
    public:  
        static int getX( ) { return x; }  
        static int getY( ) { return y; }  
};  
int A::x = 0, A::y = 0;  
...
```

How to use class A?

Concept

```
class A {  
    private:  
        static int x;  
        static int y;  
    public:  
        static int getX( ) { return x; }  
        static int getY( ) { return y; }  
};  
int A::x = 0, A::y = 0;  
...  
int a = A::getX( );
```

Questions

```
class A {  
private:  
    static int x;  
    static int y;  
    int z;  
public:  
    static int getX( ) const { return x; }  
    static int getY( ) const { return y; }  
    static int getZ( ) const { return z; }  
  
};
```

Any errors?

Answers

```
class A {  
    private:  
        static int x;  
        static int y;  
        int z;  
    public:  
        //a type qualifier is not allowed on a static member function  
        static int getX( ) const { return x; }  
        static int getY( ) const { return y; }  
        static int getZ( ) const { return z; } // z is a non-static data member.  
};  
// static members are not initialized.
```

Any errors?

```
class A {  
    private:  
        static int x;  
        static int y;  
        static int z;  
    public:  
        static int getX( ) const { return x; }  
        static int getY( ) const { return y; }  
        static int getZ( ) const { return z; }  
};  
int A::x = 0, A::y =0; A::z = 0;
```


So we have

```
class A {
```

```
private:
```

```
    static int x;
```

```
    static int y;
```

```
    static int z;
```

```
public:
```

```
    static int getX( ) { return x; }
```

```
    static int getY( ) { return y; }
```

```
    static int getZ( ) { return z; }
```

```
};
```

```
int A::x = 0, A::y = 0; A::z = 0;
```

Question

```
class A {  
public:  
    static int a;  
    int y;  
    void g0 () { a = y + 1; }  
    static void g1 () {  
        g0 ( );  
        a = y +1;  
    }  
};
```

Any errors?

Answer

```
class A {  
public:  
    static int a;  
    int y;  
    void g0 () { a = y + 1; }  
    static void g1 () {  
        g0 ( );           // cannot call a non-static method  
        a = y + 1;        // y is not a static member  
    }  
};  
int A::a = 0;              // need initialization
```

The last example 😊 is coming.....

Questions

```
class A {  
public:  
    static int x;  
    static int compute() const { return ++x;}  
  
};  
  
int main( )  
{  
    A::x = 10;  
}
```

Any errors?

Answers

```
class A {  
public:  
    static int x;  
    static int compute() { return ++x;}  
    // for a static function, there is no modifier, such as const  
};  
int A::x = 0;  
int main( )  
{  
    A::x = 10;  
    return 0;  
}
```

Question

- Implement the functions to check the answers on your own.

```
class A {
public:
    A() {
        name = "nA";
        cout << "Constructor A:" << name << endl;
    }
    A(const string &a) {
        name = a;
        cout << "Constructor A(string):" << name << endl;
    }
    A( const A &a) {
        cout << "Copy constructor A:" << a.name << endl;
        name = "Copy:" + a.name;
    }
    A &operator=(const A &a) {
        cout << "Assignment operator A" << endl;
        name = name + "=" + a.name;
        return *this;
    }
    ~A() {
        cout << "Destructor A:" << name << endl;
    }
    string name;
};
```



```
class B: public A {
public:
    B( ) {
        name = "nB";
        cout << "Constructor B:" << name << endl;    }
    B(const string &b) {
        name = b;
        cout << "Constructor B(string):" << name << endl;    }
    B( const B &b) {
        name = "Copy:" + b.name;
        cout << "Copy constructor B:" << name << endl;    }
    B &operator=(const B &b) {
        cout << "Assignment operator B" << endl;
        name = name + "=" + b.name;
        return *this;    }
    B operator+( const B &ib) {
        B b;
        b.name = b.name + "+" + ib.name;
        return b;    }
    ~B() {
        cout << "Destructor B:" << name << endl;    }
};
```

Invoke test_00. What are the outputs?

Invoke test_01. What are the outputs?

```
void f0(A a)
{
}
void test_00( ) {
    cout << "begin test_00" << endl;

    A a;
    B b, b1, b2, b3;
    f0( B() );
    b1 = b2 + b3;
    cout << "end test_00" << endl;
}
```

```
void test_01( ) {
    cout << "begin test_01" << endl;
    A a("a");
    B b("b"), b1("b1");
    B b2("b2"), b3("b3");
    f0( B("func") );
    b1 = b2 + b3;
    cout << "end test_01" << endl;
}
```