

西安电子科技大学

考试时间 120 分钟



试 题

题号	一	二	三	四	总分
分数					

1. 考试形式：闭卷； 2. 本试卷共 四 大题，满分 100 分。

班级_____学号_____姓名_____任课教师_____

Part I There is one error in each code paragraph. Find out the error and write down the error statement on your answer sheet. (20 points)

(1)	<pre>int *zPtr; int z[5] = {1, 2, 3, 4, 5}; zPtr = ++z;</pre> <p><i>zPtr = 2, z[0]</i></p>	(2)	<pre>class student { int marks; } student s1; student s2 = 2;</pre> <p><i>public: Student(int);</i></p>
(3)	<pre>int f(const int x, int y){ x += y; return x; }</pre>	(4)	<pre>class C { int& c; public: C() { c = 0; } };</pre> <p><i>引用本身无错</i></p>
(5)	<pre>namespace mySpace1 { float x; } namespace mySpace2 { int i; float x; }; using mySpace::x = 1;</pre>	(6)	<pre>template <class T> class myTemp { public: void m(); // }; void myTemp::m() { /* m's body */ }</pre> <p><i>缺少模板类参数列表</i></p>

↑ 未说明命名空间，但即使说了也不对

using mySpace1::x = 1; 不对!

using mySpace::x; 对✓

(7)	<pre> class MyClass { private: int x; public: MyClass(int val) : x(val) { } void set(int i) { x = i;} int get() {return x;} }; int main() { const MyClass foo(10); foo.set(20); return 0; } </pre>	(8)	<pre> class BC { public: BC(int a) { x = a; z = -1; } BC(int a1, int a2) { x= a1; z = a2; } private: int x, z; }; class DC : public BC { public: DC(int a) { y = a;} private: int y; }; </pre>
(9)	<pre> #include <iostream> using namespace std; int main() { try { int a = 9; throw a; } catch (...) { /* */ } catch (int k) { /* */ } return 0; } </pre>	(10)	<pre> class CArray { public: void CArray(int i) { length = i; buffer = new char[length]; } ~CArray(){delete [] buffer;} private: int length; char *buffer; }; </pre>

Part II Write the following programs' output. (30 points)

1. (6 points)

```

#include <iostream>
using namespace std;
int main() {

```

```

    int a[] = {10,30,50,40,20};

```

```

    int *p = a;

```

```

    for(int i = 0; i < 5; i++)

```

```

        cout << *p++ << "\t"

```

```

    cout << endl;

```

```

    int &r = a[2];

```

*p++ = *(p++) 地址+1

(*p)++ 数值+1

4个空格

10 30 50 40 20

10 30 55 40 20

```

    r = r + 5;
    for(int i = 0; i < 5; i++)
        cout << a[i] << "\t";
    return 1;
}

```

2. (6 points)

```
#include <iostream>
```

```
using namespace std;
```

```
void foo() {
```

```
    static int a = 1;
```

```
    int b = 0;
```

```
    cout << a++ << '\t' << ++b << endl;
```

```
}
```

```
int main() {
```

```
    for(int i = 0; i < 3; ++i)
```

```
        foo();
```

```
    return 0;
```

```
}
```

全局变量

1

1

2

1

3

1

记录

不会被记录

3. (6 points)

```
#include <iostream>
```

```
using namespace std;
```

```
class A {
```

```
    int v;
```

```
public:
```

```
    A() { cout << "A1" << endl; }
```

```
    A(int v) {
```

```
        this->v = v; cout << "A2, v=" << this->v << endl;
```

```
    }
```

```
};
```

```
int main(){
```

```
    A a1[2];
```

```
    A a2[2] = {5, 10};
```

```
    A *p = new A[2];
```

```
    return 0;
```

```
}
```

A1

A1

A1, v=5

A1, v=10

A1

A1

4. (6 points)

```
#include <iostream>
using namespace std;
class B{
public:
    virtual void foo(){ cout << "Base::foo()\n"; }
};
class C:public B{
public:
    void foo() {cout << "Derived::foo()\n"; }
};
int main() {
    C c;
    B b, &p = c;
    c.foo();
    b.foo();
    p.foo();
    return 0;
}
```

D
B
D

5. (6 points)

```
#include<iostream>
using namespace std;
class A{
    double x, y;
    static int cnt;
public:
    A(double a = 0, double b = 0) : x(a), y(b){ cnt++; }
    ~A(){ cnt--; cout << "~A(): " << cnt << endl; }
    void print(){
        cout << "Object: (" << x << ", " << y << ")t";
        cout << "number of A: " << cnt << endl;
    }
};
int A::cnt = 0;
int main(){
    A a;
    A *p = new A(1.5);
```

Object: (1.5, 0)

number of A: 2

~A(): 1

Object: (0, 0)

number of A: 1

~A(): 0

class object { }

class fruit : public object

class apple : public fruit
 {
 p->print();
 delete p;
 a.print();
 return 0;
 }

class fruit { }

class apple : public fruit { }

class grape : public fruit { }

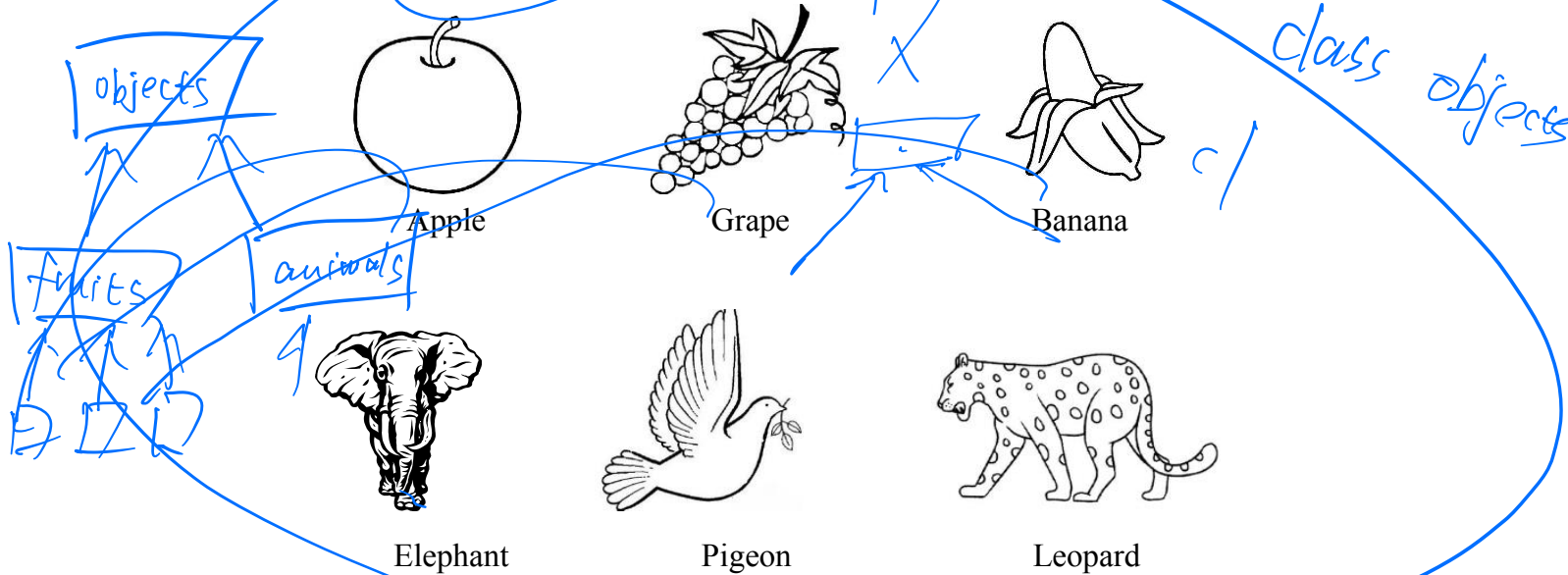
class banana : public fruit { }

class Animal { }

class elephant : public animal { }

Part III Object Oriented Analyzing and Designing (30 points)

1. From following pictures, please analyze and design the class and class Hierarchies. (15 points)



2. Please define a class **Book** and its data members and member functions according to the following description. Don't implement any member functions.

- (1) Each book has a title, one or more authors, and an international standard book number (ISBN).
- (2) Class **Book** has two constructor:
 - (a) the first should be a default constructor,
 - (b) the second should take two arguments: the title and the ISBN.
- (3) The authors can be added one by one.
- (4) For the title and ISBN of a **Book**, a "set" function and a "get" function are required. Each "set" function updates the corresponding data member to a new value through its parameters, and each "get" function returns the value of the corresponding data member to caller.

Part IV Programming (20 points)

1. Define and implement a class named **Integer** according to the main

function and the output given in comments.

```
int main()
{
    Integer a(2), b=a, c;
    a.print("a="); // a=2
    b.print("b="); // b=2
    c.print();      // 0

    c = b + 1;
    b.print("b="); // b=2
    c.print("c="); // c=3

    c = a - c;
    a.print("a="); // a=2
    c.print("c="); // c=-1

    c *= b;
    c.print("Finally, c="); // Finally, c=-2
    return 0;
}
```

2. Given a class :

```
class Sequence {
    protected: int number;
    public:
        Sequence(int v) : number(v) { }
        virtual void action(){}
        virtual int getNumber() { return number ; }
};
```

According to the main function and the output below, please define and implement class **Increment**, **Square**, and **Decrement**, they are direct derived(sub) classes of **Sequence**. Don't modify the class **Sequence**.

```
int main( )
{
    Increment inc(1); Square pow(2); Decrement dec(9);
    Sequence * ptrs[3];
    ptrs[0] = &inc;    ptrs[1] = &pow;    ptrs[2] = &dec;
```

```

for( int i = 0; i < 3; i++ ) {
    for( int k = 1; k <= 5; ++ k ) {
        cout << ptrs[i]->getNumber() << "\t";
        ptrs[i] -> action();
    }
    cout << endl;
}
return 0;
}

```

The output is:

1	2	3	4	5
2	4	16	256	65536
9	8	7	6	5

Part I There is one error in each code paragraph. Find out the error and write down the error statement on your answer sheet.

(1)	<pre> class C { int x; void setx(int a) { /* ... */ } }; void main() { C c1; c1.setx(3); } </pre>	(2)	<pre> class Student { //... public: void Student(); ~ Student(); }; </pre>
(3)	<pre> class BC { int x; public: BC(int xx = 0) { x = xx; } }; class DC : public BC { char c; DC(int x1, char c1) { x = x1; c = c1; } }; </pre>	(4)	<pre> class C { int sz; public: friend C operator+ (const C&, const C&); // ... }; C C::operator+ (const C& c1, const C& c2){ cout << c1.sz; // } </pre>
(5)	<pre> class C { public: void m() { /* ... */ } static void s() { /* ... */ } }; void main() { C c1; c1.m(); C::m(); c1.s(); C::s(); } </pre>	(6)	<pre> template <class T, int i> class Array { int sz; public: Array():sz(i) { } // }; void f(int x) { Array<int, x> ay; } </pre>

Part II Write the following programs' output.

1.

```
#include <iostream>
using namespace std;
int main() {
    int i = 1;
    while(i <= 6) {
        i++;
        if(i % 3 != 1) continue;
        else cout << i << " ";
    }
}
```

4 7

2.

```
enum TorF { F, T=48 };
int cvt (char c) { cout << "1:";
    return c; }
int cvt (int c) { cout << "2:";
    return c; }
int cvt (TorF v) { cout << "3:";
    return (v==T)? 1 : 0; }
int main() {
    cout << cvt(char(48)) << endl;
    cout << cvt(T) << endl;
    cout << cvt(!true) << endl;
    return 0;
}
```

1: 48

3: 1

2: 0

3.

```
#include <iostream>
using namespace std;
void add1(int a1) { a1++; }
void add2(int& a2) { a2++; }
void add3(int* a3) { (*a3)++; }
int main() {
    int t1=2, t2=2, *t3=&t2, t4=t2, &t5=t2;
    add1(t1); add2(t2); add3(t3);
    add3(&t4); add2(t5);
    cout << t1 << " " << t2 << " " << *t3 << " " << t4 << " " << t5 << endl;
    return 0;
}
```

2 5 5 3 5

4.

```
#include <iostream>
using namespace std;
class B {
public:
    void m() { cout << " B::m" << endl; }
```

```

        virtual void f ()      {   cout << " B::f " << endl;   }
};
class D: public B {
public:
    void m()      {   cout << " D::m" << endl;   }
    void f ()     {   cout << " D::f " << endl;   }
};
int main() {      B *p[2];
    p[0] = new D;      p[1] = new B;
    for(int i = 0; i <= 1; i++) { p[i]->m();
    return 0;
}

```

B::m
 D::f
 p[i]->f();
 B::m
 B::f

Part III Object-Oriented Analyzing and Designing (30 points)

1. From following pictures, please analyze and design the class and class Hierarchies. (15 points)

《图略》

2. (15 points)

Define a class named **Point** which can express the position of any point in a plane coordinate(坐标) system. A **Point** object contains two private data member: **x** which holds the horizontal coordinate, **y** which holds the vertical coordinate. This class should have such public operations:

- a default constructor that set the coordinate to (0,0);
- a constructor that takes two integer, which initializes the x and y;
- an overloaded operator “+=” which moves the point to another position;
- some member functions to re-set or get each of two data members.

Part IV Programming (20 points)

1. (10 points)

Define and implement a class “**MyString**” according to the main() and the output in comments.

```

int main()
{

```

```

MyString s1("0123456789"), s2(5), s3;
    s1.display();           // Output: [0123456789]
    s2.display();           // Output(5 spaces between []): [      ]
    s3.display();           // Output(no space between []): []
s3 = s1;
    s1.display();           // Output: [0123456789]
    s3.display();           // Output: [0123456789]
s2 = s1 + 3;
    s1.display();           // Output: [0123456789]
    s2.display();           // Output: [3456789]
s3 = ++++s2;
    s2.display();           // Output: [56789]
    s3.display();           // Output: [56789]
return 0;
}

```

2. (10 points) According to the main function and the output below, implement a class hierarchy with *fighter* as the base class and *Warrior* is a derived class from *fighter*.

```

int main()
{
    fighter * objs[2];
    objs[0] = new fighter("Harry");
    Warrior Stallone("Stallone", objs[0]);
    objs[1] = &Stallone;
    cout << "Test the class famliy:" << endl;
    for(int i=0; i<2; i++) {
        objs[i] -> outTitle();
        cout<< "----  " << i+1 << " ----\n";
        objs[i] -> hello();
    }
    cout << "This is the end." << endl;
    delete objs[0];
    return 0;
}

```

The output of this program is:

Test the class famliy:

We are fighters.

---- 1 ----

Harry is a fighter.

We are fighters.

---- 2 ----

Stallone is a stronger warrior
than Harry.

This is the end.

西安电子科技大学

考试时间 120 分钟

试 题



题号	一	二	三	四	总分
分数					

1. 考试形式：闭卷； 2. 本试卷共 四 大题，满分100分。

班级_____ 学号_____ 姓名_____ 任课教师_____

Part I There is one error in each code paragraph. Find out the error and write down the error statement on your answer sheet. (20 points)

(1) <pre>float* ptr = new float[20]; for (int i = 0; i < 20; i++) ptr[i] = i+2; delete ptr;</pre> <p>27</p>	(2) <pre>namespace myspace{ void do() { /* ... */ int temp; } } using namespace myspace; using namespace myspace::temp;</pre>
(3) <pre>class C{ int x; void setx(int a) { /* ... */ }; } void main() { C c1; c1.setx(3); }</pre>	(4) <pre>class Student { //... public: void Student(); ~ Student(); };</pre>
(5) <pre>class BC { int x; public: BC(int xx = 0) { x = xx; } }; class DC : public BC { char c; DC(int x1, char c1) { x = x1; c = c1; } };</pre>	(6) <pre>class C { int sz; public: friend C operator+ (const C&, const C&); // ... }; C C::operator+ (const C& c1, const C& c2){ cout << c1.sz; // }</pre> <p>62</p>
(7) <pre>class B { public:</pre>	(8) <pre>class Animal { public:</pre>

BC, ...

	<pre> B(int a=10, float y) { i=a; z=y; } private: int i; float z; }; </pre>	<pre> virtual void f()=0; }; void f(){ Animal b; } </pre>
(9)	<pre> class C { public: void m() { /* ... */ } static void s() { /* ... */ } }; void main() { C c1; c1.m(); C::m(); c1.s(); C::s(); } </pre>	(10) <pre> template <class T, int i> class Array { int sz; public: Array():sz(i) {} // }; void f(int x) { Array<int, x> ay; } </pre>

Part II Write the following programs' output. (30 points)

1. (6 points)

```

#include <iostream>
using namespace std;
void main() {
  int i = 1;
  while(i <= 15) {
    i++;
    if(i % 3 != 1) continue;

    else cout << i << " ";

  }
}

```

4 7 10 13 16

2. (6 points)

```

#include <iostream>
using namespace std;
class IntNumber{
  int value;
public:
  IntNumber(int value){this->value = value;}
  IntNumber& operator*=(int v){ value *= v; return *this;}
  friend void operator << (ostream& outf, IntNumber& n){ outf << n.value << endl;}
};
template<class T> const char* cmp(T a, T b) { return "<T>CMP"; }
const char* cmp(IntNumber a, IntNumber b) { return "<IntNumber>cmp"; }
const char* cmp(IntNumber a, int b) { return "<IntNumber, int>cmp"; }
int main() {
  IntNumber a(1), b(2);
  b *= 10;
  cout << "b=" << b;
  cout << cmp(a, b) << endl;
  cout << cmp(a, 0) << endl;
  cout << cmp(1, 0) << endl;
}

```

b=20
 <Int Number> cmp
 <Intnumber, int> cmp
 <T> cmp

```
return 0;
```

3. (6 points)

```
#include <iostream>
using namespace std;
void add1(int a1) {
```

```
    a1++;
```

```
}
```

```
void add2(int& a2) {
```

```
    a2++;
```

```
}
```

```
void add3(int* a3) {
```

```
    (*a3)++;
```

```
}
```

```
int main() {
```

```
    int t1=2, t2=2, *t3=&t2, t4=t2, &t5=t2;
```

```
    add1(t1); add2(t2); add3(t3); add3(&t4); add2(t5);
```

```
    cout<<t1<<" "<<t2<<" "<<*t3<<" "<<t4<<" "<<t5<<endl;
```

```
    return 0;
```

```
}
```

4. (6 points)

```
#include <iostream>
```

```
using namespace std;
```

```
class B {
```

```
public:
```

```
    void m() { cout << " B::m" << endl; }
```

```
};
```

```
class D: public B {
```

```
public:
```

```
    void m() { cout << " D::m" << endl; }
```

```
};
```

```
int main() {
```

```
    B *p[2];
```

```
    p[0] = new D;
```

```
    p[1] = new B;
```

```
    for(int i = 0; i <= 1; i++)
```

```
        p[i]->m();
```

```
    return 0;
```

```
}
```

5. (6 points)

```
#include <iostream>
```

```
using namespace std;
```

```
class C{
```

```
    int value;
```

```
public:
```

```
    C(int v) : value(v) { }
```

```
    bool operator< ( C& b ) {
```

```
        if( value < b.value ) return true;
```

```
        return false;
```

```
    }
```

```
    int getValue() { return value; }
```

```
};
```

```
double getMin(double a, double b) {
```

```
    cout << "getMin(double, double) is invoked!\n";
```

2 5 5 35

Bum
Bum

```

    if( a < b ) return a;
    return b;
}
C getMin(C& a, C& b) {
    cout << "getMin(C&, C&) is invoked!\t";
    if( a < b ) return a;
    return b;
}
C getMin(C a, C b) {
    cout << "getMin(C, C) is invoked!\t";
    if( a < b ) return a;
    return b;
}
int main() {
    C c1(-9), c2(90);
    cout << getMin(10, 99) << endl;
    cout << getMin(1.2, 2.3) << endl;
    cout << getMin(10, 2.3) << endl;
    cout << getMin(c1, 2).getValue() << endl;
    return 0;
}

```

get / min(double, double) --- 99
get / min(double, double) --- 2.3
get / min(double, double) --- 10
 get Min (C, C) --- 9

Part III Object-Oriented Analyzing and Designing (30 points)

1. From following pictures, please analyze and design the class and class Hierarchies. (15 points)



游泳
Swimming



篮球
Basketball



足球
Football



赛艇
Rowing



帆船
Sailing



排球
Volleyball

class 体育运动项目 { }

class 水上项目: public 体育运动项目 { }

class 陆地项目: public 体育运动项目 { }

class 游泳: public 水上项目 { }

2. (15 points)

Define a class named **Point** which can express the position of any point in a plane coordinate(坐标) system. A **Point** object contains two private data member: **x** which holds the horizontal coordinate, **y** which holds the vertical coordinate. This class should have such public operations:

- a) a default constructor that set the coordinate to (0,0);
- b) a constructor that takes two integer, which initializes the x and y;
- c) an overloaded operator “+=” which moves the point to another position;
- d) some member functions to re-set or get each of two data members.

Point & operator += (const Point & p);

Part IV Programming (20 points)

1. (10 points)

Define and implement a class “**MyString**” according to the main() and the output in comments.

```
int main()
{
    MyString s1("0123456789"), s2(5), s3;
    s1.display();           // Output: [0123456789]
    s2.display();           // Output(5 spaces between []): [   ]
    s3.display();           // Output(no space between []): []
    s3 = s1;
    s1.display();           // Output: [0123456789]
    s3.display();           // Output: [0123456789]
    s2 = s1 + 3;
    s1.display();           // Output: [0123456789]
    s2.display();           // Output: [3456789]
    s3 = ++++s2;
```



```
s2.display();    // Output: [56789]
s3.display();    // Output: [56789]
return 0;
}
```

2. (10 points)

According to the main function and the output below, implement a class hierarchy with ***fighter*** as the base class and ***Warrior*** is a derived class from ***fighter***.

```
int main()
{ fighter * objs[2];

  objs[0] = new fighter("Harry");
  Warrior Stallone("Stallone", objs[0]);
  objs[1] = &Stallone;
```

```

cout << "== Test the class famliy ==" << endl;
for(int i=0; i<2; i++) {
    objs[i] -> outTitle();
    cout << "---- " << i+1 << " ----" << endl;
    objs[i] -> hello();
}
cout << "==This is the end." << endl;
delete objs[0];
return 0;
}

```

The output of this program is:

```

== Test the class famliy ==
We are fighters.
---- 1 ----
Harry is a fighter.
We are fighters.
---- 2 ----
Stallone is a stronger warrior than Harry.
==This is the end.

```