警示

《中山大学授予学士学位工作细则》第六条

考试作弊不授予学士学位

计算机科学系　2013学年度第三学期

《　程序设计 II　》期末考试试题(A)

任课教师：吴维刚　刘聪　考试形式：　闭卷　考试时间：2小时

年级：　班别：　专业：　　　姓名：________　学号：__________成绩

1. Single Choice (20points)

Please choose the choice that best completes the statement in the question

1) Which of the following statements about object-oriented programming (OOP) is NOT correct?
   A) An object-oriented program can run faster than a procedure-oriented program with the same functionality.
   B) OOP offers many advantages: simplicity, modularity, modifiability, and so on.
   C) OOP is more efficient than procedure-oriented programming (POP) in terms of programming productivity.
   D) The key idea of OOP is to build programs using software object.

2) Which of the following statements about static data members is NOT correct?
   A) A static data member can be initialized at declaration.
   B) A static data member refers to a member whose value cannot be changed after initialized.
   C) We may access static data members before any object is created.
   D) A static data member is shared by objects of the same class.

3) Which of the following variables cannot be a member of class A?
   A) A *p　B) A a　C) A &r　D) string s

4) Which of the following statements is not a characteristic of a constructor?
   A) The name of a constructor must be the same as the class.
   B) A constructor can be overloaded.
   C) The parameters of constructor may have default values.
   D) The return type of a constructor must be "void".

5) If C is a class name, how many times is the constructor of C called in "C a, b[2], *p[2];"?
   A) 5　B) 4　C) 3　D) 2

6) If p is a pointer to an object with a member function x(), which of the following access to x() is correct?

A) *p.x B) p->x() C) *p->x() D) *p.x()

7) Which of the following is NOT a member function of a class?
A) constructor B) destructor C) friend function D) copy constructor

8) If class X is declared to be the friend of a class Y, then which of the following statements is correct?
A) Y can access the protected members of X.
B) Y becomes X’s friend automatically
C) If class Z is declared to be friend of X, Z becomes Y's friend.
D) X can access the private members of Y.

9) Which of the following declarations of constant function member is correct?
A) void print() const;
B) const void print();
C) void const print();
D) void print(const);

10) To realize polymorphism, the following two elements are necessary:
A) Multiple-inheritance and virtual function
B) Virtual inheritance and function overloading
C) Virtual function and function invocation via a pointer/reference of base class
D) Virtual inheritance and dynamic casting

11) With a base class B and its derived class D, which of the following statements has a compilation error?
A) B b(D()); B) B * b = new D(); C) B b = D(); D) B * b = D();

12) Which of the following statements about virtual function is correct?
A) Virtual function is not a member function.
B) Virtual function must be a static function.
C) A virtual function in a base class must be defined, even though it may not be used.
D) A virtual function is function without implementation.

13) What is the printout of the following statements?
```
char *p = “abcd”;
p+=2;
cout<<p;
```
A) cd B) ab C) abcd D) memory error

14) Which of the following definitions about pure virtual function is correct?
A) virtual void fun()=0;
B) void fun(int)=0;
C) virtual void fun(int);

D) virtual void fun(int){}

15) Which of the following statements about the operator "new" is NOT correct?
A) The new operator is used to allocate storage dynamically.
B) The new operator returns an object or a simple variable.
C) The delete operator should be used to destroy the variable/object created by "new".
D) The new operator is necessary to realize deep copy.

16) The constructor of a base class
A) Cannot have any parameter.
B) Need to be called explicitly.
C) Can be used to create an object of a derived class.
D) Is executed after the constructor of a derived class.

17) An exception thrown by a function
A) Cannot be caught by the catch block in the same function
B) Can be handled exactly once
C) Can only be an object of the exception classes defined by C++ standard
D) Can be a variable defined by the programmer

18) With "string s1; char s2[20]; cin>>s2;", which statement is NOT correct?
A s2 = s1;   B) s1.append(s2);   C) s1 = s2;   D)) s1 = s1+s2;

19) Which is NOT an I/O object?
A) std::cin   B) std::cout   C) std:: cerr   D) iostream

20) To output data into a file, it is necessary for you to
A) make sure the file is already there before opening it.
B) open it in binary mode.
C) know the format of the data.
D) convert the data to characters (or an object of type string).

2. Output analysis (20points, 5points for each)

Please write down the printout of the following programs.

1) A program about constructor and copy constructor

```
#include <iostream>
using namespace std;
class Data{
public:
    Data(char s, float p){ size = s; price = p;}
    void show(){
        cout << size << " " << price << endl;
    }
private:
    char size;
    float price;
};
class Desk{
private:
    int id;
    Data *data;
public:
    Desk(int i, char s, float p){
        id = i;
         data = new Data(s, p);
        cout << "constructor is called.\n";
    }
    Desk(Desk &d){
        id = d.id+1;
        data = new Data(*(d.data));
        cout << "copy constructor is called.\n";
    }
    ~Desk(){ delete data; }
    void display(){
        cout << id << endl;
        data->show();
    }
};
int main(){
    Desk a1(1, 'S', 10);
    Desk a2 = a1;
    cout << "a1="; a1.display();
    cout << "a2="; a2.display();
    return 0;
}
```

2) A program about inheritance

```
#include <iostream>
using namespace std;
class Base{
public:
    Base(int n) { x = n; }
    void squareIt(){ x = x*x; }
    void output() { cout << x << endl; }
protected:
    int x;
};
class Sample : public Base{
public:
    Sample(int n = 0): Base(n) { s = n*2; }
    void sumIt(){ s = x+s; }
    void output(){
        Base::output();
        cout << s << endl;
    }
private:
    int s;
};
int main(){
    Sample s(2);
    s.sumIt();
    s.output();
    s.squareIt();
    s.output();
}
```

3) A program about polymorphism

```
#include <iostream>
#include <string>
using namespace std;

class Animal {
public:
    Animal(char* mytype)
    {
        strcpy(type, mytype);
        cout << "Animal Constructor.\n";
    }
    virtual void show()
    {
        cout << "I am a " << type << endl;
    }
protected:
    char type[10];
};

class Cat :public Animal {
public:
    Cat(char* mytype) :Animal(mytype)
    {
        cout << "Cat Constructor.\n";
    }
    virtual void show()
    {
        Animal::show();
        cout << "Mew Mew" << endl;
    }
};

class Dog :public Animal {
public:
    Dog(char* mytype) : Animal(mytype)
    {
        cout << "Dog Constructor.\n";
    }
    virtual void show()
```

```
        {
            Animal::show();
            cout << "Wong Wong" << endl;
        }
};
void Print1(Animal* a) { a->show(); }
void Print2(Animal a) { a.show(); }
void main()
{
    Cat cat("Cat");
    Dog *dog = new Dog("Dog");
    Print1(&cat);       Print1(dog);
    Print2(cat);        Print2(*dog);
}
```

4) A program about template and exception

```
#include <iostream>
#include <vector>
using namespace std;

class EmptyStackException {};

template <typename E>
class Stack
{
 private:
   vector<E> impl;

 public:
     void push(const E & e) {
         impl.push_back(e);
    }

   E pop() {
       if (impl.size() == 0)
           throw EmptyStackException();
       E e = impl[impl.size() - 1];
       impl.pop_back();
       return e;
   }
 };

 int main() {
     Stack<double> stack;
     for (int i = 0; i < 3; ++ i) {
         stack.push(0.5 + i);
   }
   try {
       while (true) {
           cout << stack.pop() << endl;
       }
   }
```

```
  catch (EmptyStackException & ex) {
     cout << "caught: EmptyStackException" << endl;
  }
}
```

3. Error Correction (20points)

There are 5 errors in each of the following program, and totally 15 errors.
You need to find out 10 of them, which you are most confident.
If you list more than 10 errors, only the first 10 errors will be graded.

1). Class and object

```
#include <iostream>
using namespace std;

 class MyClass
 {
   public:
         MyClass(int x){
               member=x;
         }
         getMember() {
               return member;
         }
         void setMember(int m){
               member=m;
          }
         void ~MyClass(){}
    private:
          int member=0;
 };

int main(){
     MyClass obj1;
     MyClass obj2(3);
     obj1.setMember(1);
     cout<<obj2.member<<endl;
     return 0;
}
```

2). Inheritance and Template

```
#include <iostream>
using namespace std;

template<class TYPE>
```

```
class BASE
{
  public:
     void show(TYPE obj){
         cout<<obj<<endl;
     }
}

template <class TYPE, class TYPE1>
class DERIVED : : public BASE<TYPE1> {
public:
    void show(TYPE obj1, TYPE1 obj2){
              cout<<obj1<<endl;
              BASE::show(obj2);
    }
};

int main(){
        DERIVED<char *, double>    obj;
        BASE<char *, double>    *pBase = &obj;
        DERIVED<char *, double>    *pDerived = pBase;
        obj.show(“Pi is”, 3.14);
        return 0;
}
```

3). A program handling a class for string text

```
#include <iostream>
#include <cstring>
using namespace std;

const int S = 100;

class String{
private:
        char array[S];
        int size;
public:
        String(){

                cin >> array;
                while (array[size])   ++size;
        }
        String(char *t){
```

```
        array = t;
    }
    ~String(){
        delete [] array;
    }
    int getSize(){ return size; }
    String &operator += (String str){
        int count = str.getSize();

        if (size + count > S){
            cout << "not enough space.\n";
            return *this;
        }
        for (int i = 0; i < count; i++)
            array[size + i] = str.array[count];

        array[size + count] = '\0';


    }
    void show(){
        cout << array << endl;
    }
};

int main(){
    String str1, str2;
    str1 += str2;
    str1.show();
}
```

4. Concept explanation (20points)

Please explain the following concepts with concrete examples. You can choose 4 out of all the 5 questions to answer.

1) The meaning of polymorphism.

2) The meaning of "this" pointer of a class.

3) The difference between shallow copy and deep copy.

4) The similarity and difference between the string and char array.

5) The meaning of abstract class.

5. Program design (20points)

Please design a matrix class, named Matrix. The elements are stored in two dimensional arrays of size N×N, and each element is of the type int.

You need to provide functions to implement the following functionalities:

- Operator “=”, which copies the elements of the right-side matrix to the left one;
- Operator “+”, to perform addition of two matrix;
- Operator “*”, to perform multiplication of two matrix;

An example main function using the Matrix class:

```
int main()     {
       Matrix    m(10), m1(10), m2(10); // 10 is the size of the Matrix
       for(int i=0; i<10; i++)
             for(int j=0;j<10;j++){
                     int x;    cin>>x;
                     m.append(i, j, x);    //value assignment
             }
           // …
           m1=m; m2=m;
           m=m1+m2;
           m=m1*m2;
}
```

<END>