南 京 航 空 航 天 大 学

本题分数	50
得　　分	

一、 读程序，写输出结果（每题 5 分，共 50 分）。

1.

```
class IntValue{
      int value;
      static int count;
public:
     IntValue (int i = 0):value(i) { count++;}
     IntValue (char c):value(c) { count++;}
     static int GetCount (){
         return count;
     }
     int GetValue (){ return value;}
};
int IntValue::count = 0;
int main()
{
     IntValue a, b(10), c('A');
     cout<< IntValue:: GetCount ()<<endl;
     cout<<a.GetValue()<<endl;
   cout<<b.GetValue()<<endl;
   cout<<c.GetValue()<<endl;
 return 0;
}
```

2.

```
class Book{
      string name;
      float price;
public:
    Book (string n = " ", float p = 0.0):name(n), price(p){cout << " constructor"<<endl;}
    Book(const Book& in):name(in.name), price(in.price)
    {cout<<"copy constructor"<<endl;}
    ~Book( ){cout <<"~deconstructor"<<endl;}
    string GetName(){return name;}
    float GetPrice(){return price;}
};
void PrintBook(Book b){
 cout<<b.GetName()<<" "<<b.GetPrice()<<endl;
}
int main(){
 Book book1, book2("C++", 30.0);
 PrintBook(book1);
        PrintBook(book2);
 return 0;
}
```

3.

```
class Language{
public:
        Language(){cout<<"Language"<<endl;}
        ~Language(){cout<<"~Language"<<endl;}
        virtual void Fun( ) {cout <<"Base\n" ;}
};
class CPlusPlus:public Language{
public:
        CPlusPlus(){cout<<"C++"<<endl;}
        ~CPlusPlus(){cout<<"~C++"<<endl;}
        virtual void Fun() {cout <<"Derived\n";}
};
int main(){
 Language obj1;
 CPlusPlus obj2;
 Language* ptr = &obj2;
 ptr->Fun();
 ptr = &obj1;
 ptr->Fun();
 return 0;
}
```

4.

```
class Shape{
     float area;
 public:
        Shape (){cout<<”Shape”<<endl;}
        virtual    float Area() = 0;
 };
 class Rectangle: public Shape{
     int width, height;
 public:
        Rectangle (int a, int b):width(a), height(b){cout<<”Rectangle”<<endl;}
        float Area () {return width * height;}
};

class Square: public Shape{
     int width;
 public:
        Square (int r):width (r) {cout<<”Square”<<endl;}
        float Area () {return width * width;}
};

int main(){
        Rectangle r(10, 3);
            Square s(5);
        Shape* p = &r;
        cout<<p->Area()<<endl;
        p = &s;
        cout<<p->Area ()<<endl;
        return 0;
}
```

5.

```
class IntValue{
        int val;
 public:
        IntValue (int v = 0): val(v){}
        IntValue (const IntValue & in): val(in.val){}
        IntValue operator++(int a)
        {
              cout<<"suffix"<<endl;
              IntValue    temp = *this;
              val++;
              return temp;
         }
        IntValue operator++()
        {
              cout<<"prefix"<<endl;
              val++;
              return *this;
        }
        void Print() {cout<<"val: "<<val<<endl;}
};
int main(){
 IntValue v1(5), v2, v3;
 v2 = v1++;
 v3 = ++v2;
v1.Print();
 v2. Print ();
 v3. Print ();
       return 0;
}
```

6.

```
class Complex{
        float real, imag;
 public:
        Complex(float a, float b):real(a), imag(b){cout<<"constructor"<<endl;}
        Complex(const Complex& in):real(in.real), imag(in.imag){
              cout<<"copy constructor"<<endl;
        }
        ~Complex(){cout<<"deconstructor"<<endl;}
        friend void IncreaseReal(Complex& in, int val);
        friend void IncreaseImag(Complex in, int val);
        void Print(){cout<<real<<" "<<imag<<endl;}
};
void IncreaseReal(Complex& in, int val)
{
 in.real += val;
}
void IncreaseImag(Complex in, int val)
{
 in.imag += val;
}
int main()
{
 Complex* c1 = new Complex(1, 2);
 IncreaseReal(*c1, 2);
 IncreaseImag(*c1, 2);
 c1->Print();
 delete c1;
 return 0;
}
```

7.

```
template <class T >
void Swap (T& a, T& b)
{
T temp;
     temp = a;
     a = b;
     b = temp;
}
template<class T>
     class Obj{
            T val1, val2;
        public:
            Obj(T a, T b):val1(a), val2(b){ if(val1 < val2) Swap(val1, val2);}
      void Print(){cout<<val1<<endl<<val2<<endl;}
};
int main ()
{
      int x = 10, y = 20;
            string s1 = "C++", s2 = "Java";
            Swap(s1, s2);
            Swap(x, y);
            cout<<x<<" "<<y<<endl;
            cout<<s1<<" "<<s2<<endl;
            Obj<int> obj('a', 'A');
            obj.Print();
        return 0;
}
```

8.

```
class A{
 int a,b;
public:
 A( int i, int j){ a= i ; b= j ; }
int Cal () { return a + b; }
int GetA(){return a;}
int GetB(){return b;}
};
```

```
class B : public A{
 public:
        B( int i,int j):A(i,j) { }
        int Cal (){ return GetA() * GetB(); }
};
int main() {
 B b(3, 4);
       A a = b;
     cout<<a.Cal()<<" "<<b.Cal()<<endl;
 return 0;
}
```

9.

```
class DataType{
       string name;
       int size;
public:
 DataType(string a, int b):name(a), size(b){cout<<"DataType"<<endl;}

~DataType(){cout<<"~DataType"<<endl;}
};
class Int: public DataType{
     int value;
 public:
        Int(int v):DataType("int", sizeof(int)), value(v) {cout<<"Int"<<endl;}
        ~Int(){cout<<"~Int"<<endl;}
};
class IntPair{
     Int a, b;
 public:
      IntPair (int x, int y): a(x), b(y){cout<<"IntPair"<<endl;}
            ~IntPair (){cout<<"~ IntPair"<<endl;}
};
int main() {
       IntPair* ptr2 = new IntPair(2, 5);
       delete ptr2;
 return 0;
}
```

10.

```
int main()
{
 fstream dataFile;
 dataFile.open("a.txt", ios::out);
 dataFile<<"Hello NUAA ";
 dataFile.close();

 dataFile.open("a.txt", ios::in);
 char ch1;
dataFile.seekg (-2, ios::end);
dataFile.get(ch1);
 cout<<ch1<<endl;
dataFile.seekg (0, ios::beg);
 char buf[5];
 dataFile>>buf;
 cout<<buf<<endl;
 dataFile.close();
 return 0;
}
```

答题卡

一、 读程序，写输出结果（每题 5 分，共 50 分）。

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

本题分数	10
得　　分	

二、编程题（每题 10 分，共 50 分）。

1. 写一个程序依次读入文本文件 1.txt 中的每一行，将偶数行内容添加行号后写到另一个文件 2.txt 中。

2. 实现字符数组类存储一组字符变量，请编写类中未完成的成员函数，并编写主函数测试如下成员函数：拷贝构造函数、Set、Get。

```
class CharArray{
    char* data_list;      //指针
    int len;              //数组长度
public:
    CharArray (int size);//构造函数，输入数组大小
    CharArray (CharArray& in);//拷贝构造函数
    ~ CharArray ();//析构函数
    void Set(int pos, int val);//在 pos 位置放入 val
    char Get(int pos); //返回位于 pos 位置的元素
    int Size(){return len;} //返回数组大小
};
```

3．编写一个程序，实现人民币 RMB 类，该类包含三个私有数据成员 int y, j, f，分别表示元、角、分。完成构造函数、拷贝构造函数、赋值函数。用友元函数实现重载运算符 “+”，完成两个 RMB 对象矩形相加，加法原则为:

元和元相加、角和角相加、分和分相加（注意进位）

例如 RMB m1 (10,1,3)、Money m2(2 , 4, 8)，

m1 和 m2 相加后为 m3(12, 6, 1) 编写主函数测试拷贝构造函数和运算符“+”。

4．设计一个雇员类 Employee，有以下数据成员：姓名 char name[12]、工号(int 类型)收入(float 类型)，静态数据成员雇员总人数 int num。设计构造函数完成数据成员的初始化、拷贝构造函数和必要的成员函数，添加静态成员函数返回雇员总人数。编写主程序输入三个雇员，显示每个人姓名和收入，并计算平均收入。

5. 已有如下一个基类：

```
class Variable {
        int size;   //变量占据的字节数
    public:
        Variable (int s) :size(s){}
        Variable (const Variable& in):size(in.size){}
        virtual int Calculate () = 0;
        int GetSize(){return size;}
};
```

要求：

(1) 建立两个派生类 IntVariable 和 StringVariable。其中，IntVariable 类有一个私有成员 int val 存储整型变量，StringVariable 有一个私有成员 char* ptr 存储字符串。 完成类 IntVariable 和 StringVariable 的构造函数、拷贝构造函数、析构函数。

(2) 完成类 IntVariable 和 StringVariable 的函数 Calculate()，返回变量占据的字节数。

(3) 写出主函数程序，分别创建一个值为 10 的 IntVariable 对象和“HelloNUAA”的 StringVariable 对象，并通过使用基类指针调用虚函数（即运行时的多态性）输出变量占据的字节数。

一、 读程序，写输出结果（每题 5 分，共 50 分）。

1.

```
3
0
10
65
```

2.

```
constructor
constructor
copy constructor
   0
~deconstructor
copy constructor
C++ 30
~deconstructor
~deconstructor
~deconstructor
```

3.

```
  Language
 Language
 C++
 Derived
 Base
 ~C++
~Language
~Language
```

4.

```
 Shape
 Rectangle
 Shape
 Square
30
25
```

5.

```
suffix
prefix
```

```
val:7
val:5
val:7
```

6.

```
constructor
copy constructor
deconstructor
3 2
deconstructor
```

7.

```
20    10
Java C++
97
65
```

8.

```
7    12
```

9.

```
DataType
Int
DataType
Int
IntPair
~IntPair
~Int
~DataType
~Int
~DataType
```

10.

```
A
Hello
```

二、编程题（每题 10 分，共 50 分）。

1.

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

#define MAX_SIZE 256
int main()
{
    ifstream in_file("1.txt", ios::in); //1 分
    ofstream out_file("2.txt", ios::out); //1 分
    if(in_file.fail() || out_file.fail()) return -1; //1 分
    char buffer[MAX_SIZE];
    int count = 0; //1 分
    int count2 = 1;
    while(!in_file.eof()){ //1 分
            in_file.getline(buffer, MAX_SIZE); //1 分
            count++;
            if(count % 2 == 0){          //2 分
                out_file<<count2<<" "<<buffer<<endl;
                count2++;
            }
            if(in_file.fail())break;
    }
    in_file.close(); //1 分
    out_file.close();//1 分

    return 0;
}
```

2.

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

class CharArray{
    char* data_list;   //
    int len;           //数组长度
public:
    CharArray(int size);//构造函数
    CharArray(CharArray &in);//拷贝构造函数
    ~CharArray();//析构函数
    int Set(int pos, int val);
    char Get(int pos) {return data_list[pos];}
    int Size(){return len;}
};

CharArray::CharArray (int size)//构造函数  //2 分
{
    if(size <= 0) data_list = NULL;
    else{
        len = size;
        data_list = new char[size];
    }
}
CharArray::CharArray(CharArray &in)//拷贝构造函数  //3 分
{
    if(in.len > 0){
        len = in.len;
        data_list = new char[len];
        for(unsigned int i = 0;i < len;i++)
            data_list[i] = in.Get(i);
    }
}
int CharArray::Set(int pos, int val) //1 分
{
    if(0 <= pos && pos < len) {
        data_list[pos] = val;
        return 1;
    }
    return -1;
}

CharArray::~CharArray()//析构函数  //1 分
{
    if(data_list != NULL) delete[] data_list;
}
int main()                //3 分
{
    CharArray array1(5);
    unsigned int i = 0;
    for(;i < 5;i++)
        array1.Set(i, 'a');
    CharArray array2(array1);
    for(i = 0;i < array2.Size();i++)
        cout<<array2.Get(i)<<endl;
    return 0;
}
```

3.

```
#include <iostream>
using namespace std;
class RMB{
    int y, j, f;
    public:
        RMB(int a, int b, int c):y(a), j(b), f(c){} //1 分
         RMB(const RMB& in):y(in.y), j(in.j), f(in.f){} //1 分
         RMB& operator=(const RMB& in) //2 分
        {
            y = in.y;
            j = in.j;
            f = in.f;
            return *this;
        }
        friend RMB operator+(RMB& in1, RMB& in2);
        void Print() //1 分
        {
            cout<<y<<" "<<j<<" "<<f<<endl;
        }
};
RMB operator+(RMB& in1, RMB& in2) //3 分
{
    int y = in1.y + in2.y;
    int j = in1.j + in2.j;
    int f = in1.f + in2.f;
    if(f > 10){
        j++;
        f = f % 10;
    }
    if(j > 10){
        y++;
        j = j % 10;
    }
    return RMB(y, j, f);
}
int main() //2 分
{
    RMB m1(10, 1, 3);
    RMB m2(2, 4, 8);
    RMB m3 = m1 + m2;
    m3.Print();
    return 0;
}
```

4.

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

class Employee{
    char name[12];
    int employee_id;
    float salary;
    static int num;
public:
    Employee(char* p, int id, float f):employee_id(id), salary(f)    //2 分
    {
        strcpy(name, p);
        num++;
    }
    Employee(const Employee& in)   //2 分
    {
        strcpy(name, in.name);
        employee_id = in.employee_id;
        salary = in.salary;
    }
    static int GetNum(){return num;}       //2 分
    float GetSalary(){return salary;}
    char* GetName(){return name;}
};
int Employee::num = 0;
int main()
```

```
{
    Employee e1("Tommy", 1, 2000);     //2 分
    Employee e2("Bobby", 2, 3000);
    Employee e3("Mike", 3, 2300);

    cout<<e1.GetName()<<" "<<e1.GetSalary()<<endl;    //1 分
    cout<<e2.GetName()<<" "<<e2.GetSalary()<<endl;
    cout<<e3.GetName()<<" "<<e3.GetSalary()<<endl;

    cout<<(e1.GetSalary() + e2.GetSalary() + e3.GetSalary())/ Employee::GetNum()
<<endl;   //1 分

    return 0;
}
```

5.

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

class Variable {
        int size;       //变量占据的字节数
    public:
        Variable (int s) :size(s){}
        Variable (const Variable& in):size(in.size){}
        virtual int Calculate () = 0;
        int GetSize(){return size;}
};

class IntVariable:public Variable{                //3 分
    int val;
public:
    IntVariable(int in):Variable(sizeof(in)),val(in){}
    IntVariable(const IntVariable& in):Variable(in), val(in.val){}
    int Calculate(){return sizeof(int);}
    ~IntVariable(){}
};

class StringVariable:public Variable{        //4 分
    char* ptr;
public:
    StringVariable(char* p):Variable(strlen(p)){
        ptr = new char[strlen(p) + 1];
        strcpy(ptr,p);
    }
    StringVariable(const StringVariable& in):Variable(in){
        if(ptr == NULL){
            ptr = new char[GetSize()];
            strcpy(ptr,in.ptr);
        }else{
            delete[] ptr;
            ptr = new char[GetSize()];
            strcpy(ptr,in.ptr);
        }
    }
    ~StringVariable(){
        if(ptr != NULL) delete[] ptr;
    }
    int Calculate(){return strlen(ptr) + 1;}
};

int main()                        //3 分
{
        IntVariable a(10);
        StringVariable b("Hello NUAA");
        Variable* p1 = &a;
        Variable* p2 = &b;
        cout<<p1->Calculate()<<" "<<p2->Calculate()<<endl;
        return 0;
}
```