南 京 航 空 航 天 大 学

二〇一九 ～ 二〇二〇学年 第 2 学期 **《C++语言程序设计》考试试题**

考试日期：2020 年 6 月 日　　试卷类型：B　　试卷代号：

班号			学号			姓名				
题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

注意：将阅读程序的答案写在答题卡上。最后**所有**的试卷都要上交给监考老师，**若不交全考卷，将以弃考处理**。

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   Language {
      string name;
      static   int   count ;
public :
      Language (string in):name(in){cout<<name<<endl; count++ ; }
      static   int   GetCount( ){return   count ;}
} ;
int    Language:: count=0 ;
int    main( ) {
      Language c("C++");
      cout<<Language ::GetCount()<<endl;
      Language p("Python");
      cout<< Language::GetCount()<<endl;
      return 0;
}
```

2.

```
class Book{
     int id;
     float price;
public:
```

```
   Book (int a = 0, float p = 0.0):id(a), price(p){cout << " constructor"<<endl;}
   Book(const Book& in):id(in.id), price(in.price){cout<<"copy constructor"<<endl;}
   ~Book( ){cout <<"bye"<<endl;}
};
int main()
{
      Book book1, book2(2, 3.0);
      Book book3 = book1;
      return 0;
}
```

3.

```
class Programming{
public:
            Programming (){cout<<"Programming"<<endl;}
            ~ Programming (){cout<<"~ Programming "<<endl;}
            void Print(){cout<<”Print()”<<endl;}
            virtual void Fun( ) {cout <<"Fun()\n" ;}
};
class OOProgramming:public Programming{
public:
            OOProgramming (){cout<<"OOProgramming"<<endl;}
            ~ OOProgramming (){cout<<"~OOProgramming"<<endl;}
            void Print(){cout<<”PrintOO()”<<endl;}
            virtual void Fun() {cout <<"FunOO()\n";}
};
int main(){
      Programming obj1;
      OOProgramming obj2;
      Programming * ptr = &obj2;
      ptr->Print();
      ptr->Fun();
      return 0;
}
```

4.

```
class Course{
      public:
            Course(){}
            virtual void Show () = 0;
      };
```

```
class AI_Course: public Course{
    int id;
    public:
        AI_Course (int i):id(i){}
        void Show () {cout <<id<<endl;}
    };
class MachineLearning: public AI_Course {
        string name;
    public:
        MachineLearning (int i, string in): AI_Course (i),name(in){}
        void Show() { cout<<name<<endl;}
    };

int main(){
        MachineLearning c1(1, "supervised learning");
        MachineLearning c2(2, "neural networks");
        Course* p = &c1;
        p->Show();
        p = &c2;
        p->Show();
        AI_Course s = c2;
        p = &s;
        p->Show();
        return 0;
    }
```

5.

```
class Letter{
        char val;
    public:
        Letter (char v): val(v){}
        Letter (int v): val(v){}
        Letter (const Letter & in): val(in.val){}
       Letter operator++(int a)
     {
        cout<<"suffix"<<endl;
        Letter temp = *this;
        val++;
        return temp;
```

```
    }
    Letter operator++()
    {
        cout<<"prefix"<<endl;
        val++;
        return *this;
    }
    void Print() {cout<<"val: "<<val<<endl;}
};
int main(){
    Letter v1('A');
    Letter v2 = v1++;
    Letter v3 = ++v2;
     v1.Print();
    v2. Print ();
    v3. Print ();
     return 0;
}
```

6.

```
class Vector{
        int v[3];
    public:
        Vector(int a, int b, int c){v[0] = a; v[1] = b; v[2] = c;}
        friend Vector operator+ (Vector & c1, Vector & c2);
        void Print(){cout<<v[0]<< "    "<<v[1]<< "    "<<v[2]<<endl;}
};
Vector operator+ (Vector & v1, Vector & v2)
{
    Vector v(v1.v[0] + v2.v[0], v1.v[1]+ v2.v[1], v1.v[2] + v2.v[2]);
     return v;
}
int main()
{
   Vector v1(1, 2, 3), v2(4, 5, 6);
  Vector v3 = v1 + v2;
    v3.Print();
    return 0;
}
```

7.

```
template <class T >
T Plus (T a, T b)
{
      return a + b;
}
template<class T>
     class Obj{
            T val1, val2;
            public:
            Obj(T a, T b):val1(a), valu2(b){}
           void Print(){cout<<Plus(val1, val2)<<endl;}
};
int main ()
{
            Obj<int> obj1(10, 10);
            obj1.Print();
            Obj<string> obj2("C++", "Progamming");
            obj2.Print();
            return 0;
}
```

8.

```
class Employee{
string name;
     string department;
public:
      Employee(string s1, string s2):name(s1), department(s2){
       cout<<name<<" "<<department<<endl;}
       void Print(){cout<<" Employee"<<endl;}
};
class Manager: public Employee{
     int level;
       public:
            Manager(string s1, string s2, int l):Employee(s1, s2), level(l){}
            void Print(){cout<<level<<endl;}
};
int main() {
  Employee e("Mike", "CS");
  Manager m("John", "CS", 2);
    Employee* p = &e;
    p->Print();
    p = &m;
```

```
   p->Print();
   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;}
};
int main() {
       Int* ptr = new Int(2);
       delete ptr;
      return 0;
}

10.
int main()
{
      fstream dataFile;
      dataFile.open("a.txt", ios::out);
      dataFile<<"C++ Programming";
      dataFile.close();
      dataFile.open("a.txt", ios::in);
      char ch1;
       dataFile.seekg (2, ios::beg);
       dataFile.get(ch1);
      cout<<ch1<<endl;
     dataFile.seekg (2, ios::cur);
      dataFile.get(ch1);
      cout<<ch1<<" "<<dataFile.tellg()<<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 a.txt 中的每一行，将每行中的小写字母转换成大写并添加行号后写到另一个文件 b.txt 中。

2. 实现整型数组类存储一组整型变量，请编写未完成的构造函数、拷贝构造函数、析构函数和 Set(int index, int pos)，编写主函数测试。

```
class  IntArray{
    int* data;     //指针
    int len;              //数组长度
public:
    IntArray (int size);//构造函数，输入数组大小
    IntArray （IntArray& in）;//拷贝构造函数
    ~ IntArray （）;//析构函数
   void Set(int index, int val); //给位置 index 赋值为 val
    int Size(){return len;} //返回数组大小
};
```

3. 已知类 Complex 定义如下：

```
class  Complex{
       double real, imag;
public :
};
```

①设计并完成构造函数、拷贝构造函数和 Print()函数（输出实部和虚部内容）。

②通过成员函数完成 Complex 类的+和-运算符重载（real 和 imag 分别相加和相减）

③编写 main 函数测试运算符函数。

4. 设计一个图书类 Book, 有以下数据成员：书名 char name[12]、编号(int 类型)和价格 (float 类型), 静态数据成员图书总数 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图书总数。编写主程序输入三本书: (“C++”, 1, 45.2), (“C”, 2, 40.0), (“Math”, 3, 50.0))，显示每本书的名称和价格，以及图书数目总数。

5. shape类是一个表示形状的抽象类，Area()为求面积的函数。

```
class shape{
public:
    virtual float Area( )=0;
};
```

①从shape类派生长方形类(rectangle)，设计并完成该类的成员变量（长和宽）、构造函数和Area()函数
②从shape类派生正方形类(square)，设计该类的成员变量、构造函数和Area()函数
③实现main函数，通过shape指针调用rectangle和square类的Area()函数输出面积。

南 京 航 空 航 天 大 学

二〇一九 ～ 二〇二〇 学年　　第 2 学期

课程名称：《C++语言程序设计》参考答案及评分标准

命题教师：　　试卷类型：　B　　试卷代号：

一、 读程序，写输出结果（每题 5 分，共 50 分）。

1.

```
C++
1
Python
2
```

2.

```
constructor
constructor
copy constructor
bye
bye
bye
```

3.

```
Programming
Programming
OOProgramming
Print()
FunOO()
~OOProgramming
~Programming
~Programming
```

4.

```
supervised learning
neural networks
2
```

5.

```
suffix
```

prefix
val: B
val: B
val: B

6.

5 7 9

7.

20
C++Programming

8.

Mike CS
John CS
Employee
Employee

9. DataType
Int
~Int
~DataType

10. +
r 6

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()
```

```
{
    fstream in_file("a.txt", ios::in); //1 分
    ofstream out_file("b.txt", ios::out); //1 分
    if(in_file.fail() || out_file.fail()) return -1; //1 分
    char buffer[MAX_SIZE];
    int count1 = 0; //1 分
    int count2 = 0;
    while(!in_file.eof()){ //1 分
            in_file.getline(buffer, MAX_SIZE); //1 分
            count1++;

                for(int i = 0; buffer[i] != '\0';i++){    //3 分
                   buffer[i] = toupper(buffer[i]);
                }
            out_file<<count1<<" "<<buffer<<endl;
            if(in_file.fail())break;
    }
    in_file.close(); //1 分
    out_file.close();
    return 0;
}


2.
#include<iostream>
using namespace std;

class IntArray{
    int* data;     //
    int len;              //数组长度
public:
    IntArray(int size);//构造函数
    IntArray(IntArray &in);//拷贝构造函数
    ~IntArray();//析构函数
    void Set(int index, int val); //给位置 index 赋值为 val
    int Size(){return len;}
};

IntArray::IntArray (int size)//构造函数  //2 分
{
    if(size <= 0) data = NULL;
```

```
    else{
        len = size;
        data = new int[size];
    }
}
IntArray::IntArray(IntArray &in)//拷贝构造函数  //2 分
{
    if(in.len > 0){
        len = in.len;
        data = new int[len];
        for(unsigned int i = 0;i < len;i++)
            data[i] = in.data[i];
    }
}

void IntArray::Set(int index, int val)   //2 分
{
    if(0 <= val && val < len) data[index] = val;
}


IntArray::~IntArray()//析构函数  //2 分
{
    if(data != NULL) delete[] data;
}
int main()                  //2 分
{
    IntArray array1(5);
    unsigned int i = 0;
    for(;i < 5;i++)
        array1.Set(i, i);
    return 0;
}

3.
#include <iostream>
using namespace std;

class   Complex{
```

```
    double real, imag;
    public :
    Complex(double d1, double d2):real(d1), imag(d2){} //1 分
    Complex(const Complex& in):real(in.real), imag(in.imag){} //1 分
    Complex& operator+(Complex& in) //2 分
    {
        Complex c(real + in.real, imag + in.imag);
        return c;
    }
    Complex& operator-(Complex& in) //2 分
    {
        Complex c(real - in.real, imag - in.imag);
        return c;
    }
    void Print(){cout<<real<<" "<<imag<<endl;} //1 分
};

int main() //3 分
{
    Complex c1(1, 2), c2(3, 4);
    Complex c3 = c1 + c2;
    Complex c4 = c2 - c1;
    c3.Print();
    c4.Print();
}

4.
#include <iostream>
#include <string.h>
using namespace std;

class Book{
    char name[12];
    int id;
    float price;
    static int num;
public:
    Book(char* p, int id, float f):id(id), price(f)     //2 分
    {
        strcpy(name, p);
        num++;
    }
```

```
    Book(const Book& in)    //2 分
    {
        strcpy(name, in.name);
        id = in.id;
        price = in.price;
    }
    static int GetNum(){return num;}        //2 分
    float GetPrice(){return price;}
    char* GetName(){return name;}
};
int Book::num = 0; //1 分
int main()
{
    Book b1("C++", 1, 45.2);      //1 分
    Book b2("C", 2, 40);
    Book b3("Math", 3, 50);

    cout<<b1.GetName()<<" "<<b1.GetPrice()<<endl;    //1 分
    cout<<b2.GetName()<<" "<<b2.GetPrice()<<endl;
    cout<<b3.GetName()<<" "<<b3.GetPrice()<<endl;
     cout<<Book::GetNum()<<endl;    //1 分
    return 0;
}
```

5.

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

class shape{
public:
    virtual float Area()=0;
};


class rectangle:public shape{ //4 分
public:
    double w, l;
    rectangle(double a, double b):w(a), l(b){}
    float Area(){
        return w*l;
```

```
    }
};

class square:public shape{ // 4 分
    public:
        double a;
        square(double in):a(in){}
        float Area(){ return a * a;}
};
int main() //2 分
{
    rectangle r(3, 5);
    square s(4);
    shape* ptr = &r;
    cout<<"rectangle area "<<ptr->Area()<<endl;
    ptr = &s;
    cout<<"square ara "<<ptr->Area();
}
```