南京师范大学 2020-2021 学年 第 2 学期

计电学院计算机/人工智能专业 2020 年级《高级语言程序设计》

课程期末试卷（B）

班 级：_____ 任课教师：_____

学 号：_____ 姓 名：_____

题号	一	二	三	总分
得分				

（考生可以携带含手写复习内容的 1 张 A4 大小纸张进行考试，所有答案必须写在答题纸内，否则不能得分！）

得分	

一、选择题（每题 2 分，共 30 分）

1. 下面关于类和对象的描述，正确的是（ ）。

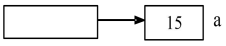
A. 对象之间不可以相互赋值

B. 一个类至少要定义一个对象

C. 对象不能用作数组元素

D. 一个对象能用作另一个对象的成员

2. 若要建立如下的指向关系，错误的语句组是（ ）。



A. int *p=&a, a; *p=15;

B. int a,*p=&a; *p=15;

C. int *p,a=15;p=&a;

D. int *p, a; p=&a;a=15;

3. 关于 new 运算符的下列描述中，( )是错的。

A. 它可以用来动态创建对象和对象数组

B. 使用它创建的对象或对象数组可以使用运算符 delete 删除

C. 使用它创建对象时要调用构造函数

D. 由 new 分配的内存空间是不连续的

4. 不能作为函数重载判断依据的是（　　）

A. 参数个数　　B. 参数类型　C. 函数名字　D. 返回类型

5. 下述静态数据成员的特征中，（　　）是错误的。

A. 说明静态数据成员时前边要加修饰符 static

B. 静态数据成员要在类体外进行初始化

C. 引用静态数据成员时，要在静态数据成员名前加<类名>和作用域运算符

D. 静态数据成员不是该类的对象所共用的。

6. 对于公有继承的派生类，若其成员函数可以直接访问基类的某个成员，说明该基类成员的访问权限是（　　）。

A. 公有或私有　　B. 公有或保护或私有

C. 保护或私有　　D. 公有或保护

7. 通过运算符重载，可以改变运算符原有的(　　)。

A. 操作数类型　　B. 操作数个数　　C. 优先级　　D. 结合性

8. 对于任意一个类，析构函数的个数最多为(　　)

A. 0　　B. 1　　C. 2　　D. 3

9. 如果类 B 被说明成类 A 的友元，则（　　）。

A. 类 A 的成员即类 B 的成员

B. 类 B 的成员函数可以访问类 A 的所有成员

C. 类 B 的成员即类 A 的成员

D. 类 A 也是类 B 的友元

10. 在下列各种符号常量的定义中，错误的定义是(　　)。

A. const float f=10.6;　　B. const int M=20;

C. const char ch;　　D. const bool mark=true;

11. 假定 AB 为一个类，px 为指向该类动态对象数组的指针，该数组长度为 n ，则执行“delete [ ] px;”语句时，自动调用该类析构函数的次数为(　　)。

A. 0　　B. 1　　C. n　　D. n+1

12. 在表达式 x+y*z 中，+是作为成员函数重载的运算符，*是作为非成员函数重载的运算符。下列叙述中正确的是(　　)。

A. operator+有一个参数，operator*有两个参数

B. operator+有两个参数，operator*有一个参数

C. operator+有两个参数，operator*有两个参数

D. operator+有一个参数，operator*有一个参数

13. 通过一个基类类型的(　　)调用虚函数时，采用动态绑定。

A. 派生类名　　B. 对象　　C. 成员名限定　　D. 指针

14. 关于虚函数，下列表述正确的是(　　)。

A. 虚函数不得声明为另一个类的友元函数

B. 如果在派生类中重定义虚函数时没有了保留字 virtual，则该重定义函数不是虚函数

C. 派生类必须重新定义基类的虚函数

D. 虚函数不能声明为静态函数

15. 下面是类 Shape 的定义：

```
class Shape
{
public:
    virtual void Draw()=0;
};
class Rect:public Shape
{
    int w,h;
};
```

下列描述中，正确的是(　　)。

A. 类 Shape 是虚基类

B. 类 Rect 是抽象类

C. 类 Shape 中的 Draw 函数声明有误

D. 语句“Rect r;”能够建立 Rect 的一个对象 s

得分	

二、程序分析题(每小题 5 分,共 40 分)

1. 下面程序的输出结果是:

```
#include <iostream>
using namespace std;
class A
{
    int a;
public:
    A(int aa=0):a(aa) { cout << a <<"  "; }
    ~A( ) {cout<<"Over;a="<<a<<endl;}
};
void main( )
{
   A  x[2]={1,2}, y;
   cout<<endl;
}
```

2. 下面程序的输出结果是:

```
#include <iostream>
using namespace std;
int &max( int &x,  int &y)
{
   return (x>y ? x : y ) ;
}
void main()
{
    int n=2, m=10;
    max(n, m) -- ;
    cout << "n=" << n << " , m=" << m << endl ;
}
```

3. 写出下面程序运行结果:

```
#include <iostream>
using namespace std;
class  Ourobj
{
    int  i, s ;
    static  int  k ;
public:
    Ourobj ( )
    {
```

```
        s = 0;
        for ( i =1 ; i <= 2; i ++ ) { s += i; k++; }
     }
    void display ( )
    {
        cout << "i=" << i<< "  k=" << k <<"  s=" << s << endl;
    }
 };
int  Ourobj::k = 0;
void  main ( )
{
   Ourobj  aa;
   aa.display ( );
   Ourobj  bb;
   bb.display( );
}
```

4.分析以下程序的执行结果

```
#include<iostream>
using namespace std;
class A
{
 public:
    int n;
};
class B:public A{};
class C:public A{};
class D:public B,public C
{
public:
    int getn(){return B::n;}
};
void main()
{
    D d;
    d.B::n=10;
    d.C::n=20;
    cout<<d.getn()<<endl;
    cout<<d.B::n<<","<<d.C::n<<endl;
}
```

5. 写出下面程序运行结果:

```
#include <iostream>
using namespace std;
class Date
{
public:
    Date(int m = 1, int y = 0) :month(m), year(y) {}
    void Print() { cout << month << "/" << year << endl; }
    friend Date  operator+(const Date&d1, const Date&d2);
private:
    int month, year;
};
Date  operator +(const Date& d1, const Date&d2)
{
    int year, month;
    year = d1.year + d2.year;
    month = d1.month + d2.month;
    year += (month - 1) / 12;
    month = (month - 1) % 12 + 1;
    return Date(month, year);
}
void main()
{
    Date d1(3, 2020), d2, d3(10);
    d2 = d3 + d1;
    d2.Print();
}
```

6. 写出程序运行结果:

```
#include<iostream>
using namespace std;
class MyClass
{
public:
    MyClass(int i = 0) { cout << 1; }
    MyClass(MyClass&x) { cout << 2; }
    MyClass& operator =(const MyClass&x) { cout << 3; return
*this; }
    ~MyClass() { cout << 4; }
};
```

```
int main()
{
    MyClass obj1(1), obj2(obj1), obj3=obj2;
    return 0;
}
```

7. 写出程序运行结果:

```
#include<iostream>
using namespace std;
class B1
{
public:
    B1(int i) { cout << "constructing B1" << i << endl; }
    ~B1() { cout << "destructing B1" << endl; }
};
class B2 {
public:
    B2() { cout << "constructing B3" << endl; }
    ~B2() { cout << "destructing B3" << endl; }
};
class C :public B2, virtual public B1 {
    int j;
public:
    C(int a, int b, int c) :B1(a), memberB1(b), j(c) {}
private:
    B1 memberB1;
    B2 memberB2;
};
int main()
{
    C obj(1, 2, 3);
}
```

8. 写出程序运行结果:

```
#include<iostream>
using namespace std;
class Base
{
public:
    Base(int i = 0) : x(i) {}
```

```
    virtual int sum() { return x; }
private:
    int x;
};
class Derived : public Base
{
public:
    Derived(int i = 0, int j = 0) :Base(i),y(j) { }
    int sum()
    {
        return Base::sum() + y;
    }
private:
    int y;
};
void Call(Base &pb)
{
    cout << "sum=" << pb.sum() << endl;
}
void main()
{
    Base b(10);
    Derived d(20, 40);
    Call(b);
    Call(d);
}
```

得分	

三、编程题：（共 30 分）

1.（10 分）编写完整程序，在 n(n=3)个学生中求出的最高分和最低分及姓名，下面给出 student 类声明和 main 函数。

student 类声明部分为：

```
#include <iostream >
#include <string>
using namespace std;
class student
{
    char name [20] ;   //姓名
    int deg;           //得分
public:
    student(char na [] ="",int d=0);
    char * getname();
```

```
    friend int compare(student &s1,student &s2);
    int getdeg();
};
void main()
{
    student st [] ={student("王强",74),
        student("李刚",68),
        student("张雪",84)};
    int i=0,min=0,max=0;
    for(i=1;i<3;i++)
    {
        if(compare(st [max] ,st [i] )==-1)
            max=i;
        if(compare(st [min] ,st [i] )==1)
            min=i;
    }
cout<<"最高分："<<st[max].getdeg()<<"姓名:"<<st[max].getname()<<endl;
cout<<"最低分:"<< st[min].getdeg()<<"姓名:"<<st[min].getname()<<endl;
}
```

答案：

2.(10 分)写一个程序，已经给定一个抽象类 Shape,由它派生 3 个类：Square(正方形)、Trapezoid（梯形）和 Triangle 三角形。用虚函数分别计算几种图形面积、并求它们的和。同时下面也给定 main 函数。

```
    #include <iostream>
    using namespace std;
    class Shape
    {
    public:
        virtual double area()const=0;
    };
    void main()
    {
        Shape *p [3] ;
        Square se(5); //边长
        Trapezoid td(2,5,4);  //上底、下底、高
        Triangle te(5,8);  //底、高
        p [0] =&se;
        p [1] =&td;
        p [2] =&te;
        double da=0;
        for(int i=0;i<3;i++)
```

```
        {
            da+=p [i] ->area();
        }
        cout<<"总面积是："<<da<<endl;
    }
```

答案：

3.（10 分）为复数类（Complex）增加重载的运算符+、+=、前置和后置自增++。设++为实部和虚部各自增一,并在下面 main 函数中测试。

答案：