2019-2020-2面向对象程序设计期末

###### ****一、 单选题（共10题，20分）****

**1、运算符重载是对已有的运算符赋予多重含义，因此(        )。**

**A、 可以对基本类型(如int类型)的数据，重新定义" + "运算符的含义**

**B、 可以改变一个已有运算符的优先级和操作数个数**

**C、 只能重载C+ +中已经有的运算符,不能定义新运算符**

**D、 C+ +中已经有的所有运算符都可以重载**

**正确答案： C**

**解析：**

**2、由于常对象不能被更新，因此(      )。**

**A、 通过常对象只能调用它的常成员函数**

**B、 通过常对象只能调用静态成员函数**

**C、 常对象的成员都是常成员**

**D、 通过常对象可以调用任何不改变对象值的成员函数**

**正确答案： A**

**解析：**

**3、下列语句中，错误的是(        )。**

**A、 const int buffer=64;**

**B、 const int temp;**

**C、 const double \*point;**

**D、 const double \*rt= new double(5.5);**

**正确答案： B**

**解析：**

**4、有如下程序:**

**int x=3;**

**do {**

**x-=2;**

**cout<<x;**

**} while(!(-x));**

**执行这个程序的输出结果是(                  )。**

**A、 1**

**B、 30**

**C、 1-2**

**D、 死循环**

**正确答案： A**

**解析：**

**5、关于静态成员的描述中，(         )是错误的。**

**A、 静态成员可分为静态数据成员和静态成员函数。**

**B、 静态数据成员定义 后必须在类体内进行初始化。**

**C、 静态数据成员初始化不使用其构造函数。**

**D、 静态成员 函数中不能直接引用非静态成员。**

**正确答案： B**

**解析：**

**6、假设OneClass为一个类，则该类的拷贝初始化构造函数的声明语句为(             )。**

**A、 OneClass (OneClass p) ;**

**B、 OneClass & (OneClass p):**

**C、 OneClass (OneClass & p);**

**D、 OneClass (OneClass \*p) ;**

**正确答案： C**

**解析：**

**7、有如下两个类定义:**

**class XX{**

**private: double x1;**

**protected: double x2;**

**public: double x3; }**

**class YY: protected XX{**

**private: double y1;**

**protected: double y2;**

**public: double y3}**

**在派生类YY中的保护成员的个数是(                      )**

**A、 1**

**B、 2**

**C、 3**

**D、 4**

**正确答案： C**

**解析：**

**8、有如下类定义:**

**class Base{**

**public:**

**void fun(){cout<<"Base: fun"<<endl;}**

**}**

**class Derived: public Base{**

**public:**

**void fun(){**

**cout<<"Derived:: fun"<<endl;}**

**}**

**如何在派生类中调用基类中的fun()函数，正确的是（     ）。**

**A、 Base. fun();**

**B、 Base->fun();**

**C、 Base:: fun():**

**D、 fun();**

**正确答案： C**

**解析：**

**9、有如下类定义:**

**class MyClass {**

**public:**

**MyClass(double d=0.0):val(d) { }**

**;   //类型转换运算符double的定义**

**private:**

**double val;**

**};**

**若要使语句序列**

**MyClass  x(2.1);**

**cout< <double(x);**

**能够正常运行，横线处的语句应为(                   )。**

**A、 operator double( ) const { return val; }**

**B、 friend operator double( ) const { return val; }**

**C、 double operator( ) const { return val; }**

**D、 friend double operator( )const { return val; }**

**正确答案： A**

**解析：**

**10、ClassA是一个类，且有如下语句序列**

**ClassA c1,\*c2;**

**ClassA \*c3= new MyClass;**

**ClassA &c4=c1;**

**则ClassA的构造函数被调用的次数是(  B  )**

**A、 1**

**B、 2**

**C、 3**

**D、 5**

**正确答案： B**

**解析：**

###### ****二、 填空题（共6题，50分）****

**1、二、程序完成题：**

**1. 类的构造函数与析构函数**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**class Employee**

**{**

**public:**

**// 构造函数：职工基本信息的初始化**

**Employee(char \*name1, bool  sex1, int  age1, int  wage1, double  salary1);**

**; // 第1空   析构函数：释放系统资源**

**void changeSalary(double \_salary);  // 修改工资**

**void print();// 显示职工信息**

**private:**

**char \*name;          // 职工姓名**

**bool sex;            // 职工性别**

**int age;             // 职工年龄**

**int wage;            // 职工工龄**

**double salary;       // 职工工资**

**};**

**// 成员函数实现**

**Employee::Employee(char \*name1, bool  sex1, int  age1, int  wage1, double  salary1)**

**{**

**int len = strlen(name1);**

**; //  第2空 在堆上开辟数组空间**

**strcpy(name, name1);**

**sex =                             ;        // 第3空**

**age =                            ;      // 第4空**

**wage =                         ;    // 第5空**

**salary =                        ;    // 第6空**

**}**

**Employee::~Employee()**

**{**

**if(name!=NULL)**

**;    // 第7空 释放数组空间**

**}**

**void Employee::changeSalary(double \_salary) // 修改工资**

**{**

**;       // 第8空**

**}**

**void Employee::print()// 显示职工信息**

**{**

**cout<<"姓名: "<< name<<endl;**

**cout<<"性别: "<< (sex?"男":"女") <<endl;**

**cout<<"年龄: "<<age<<endl;**

**cout<<"工龄: "<<wage<<endl;**

**cout<<"工资: "<<salary<<endl;**

**}**

**int main()**

**{**

**Employee em("张三",true,30,0,3000);  // 类对象定义, 调用构造函数进行初始化**

**em.print();**

**return 0;**

**}**

**正确答案：   
第1空:**

**~Employee()**

**第2空:**

**name = new char[len+1]**

**第3空:**

**sex1**

**第4空:**

**age1**

**第5空:**

**wage1**

**第6空:**

**salary1**

**第7空:**

**delete [] name**

**第8空:**

**salary = \_salary**

**解析：**

**2、二、程序完成题：**

**2. 友元函数**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**class Point  // Point 类定义**

**{**

**public:**

**Point(double  x1 = 0.0, double  y1 = 0.0);          // 带默认参数的构造函数**

**Point(const Point &point);        // 拷贝构造函数**

**void print() const;      // 打印点的坐标信息，定义为const成员函数**

**friend double pdistance(const Point &p1, const Point &p2); // 友元函数声明**

**friend Point midpoint(const Point &p1, const Point &p2);   // 友元函数声明**

**protected:**

**double x;**

**double y;**

**};**

**// Point 类成员函数实现**

**//第1空  Point构造函数**

**{**

**x = x1;**

**y = y1;**

**}**

**Point::Point(const Point &point)**

**{**

**x =                        ;           //第2空**

**y =                        ;           //第3空**

**}**

**void Point::print() const**

**{**

**cout<<                                       <<endl;         //第4空输出格式：(x,y)**

**}**

**// 普通友元函数定义，计算两个Point对象的距离**

**double pdistance(const Point &p1, const Point &p2)**

**{**

**return                         ((p1.x-p2.x)\*(p1.x-p2.x)+(p1.y-p2.y)\*(p1.y-p2.y));     //第5空**

**}**

**// 普通友元函数定义，计算两个Point对象的中点**

**Point midpoint(const Point &p1, const Point &p2)**

**{**

**Point mid;**

**mid.x=                                               ;       //第6空**

**mid.y=                                               ;        //第7空**

**return mid;**

**}**

**int main()**

**{**

**Point p1(1.0, 2.0);**

**Point p2(4.0, 6.0);**

**cout<<"距离: "<<pdistance(p1, p2)<<endl;**

**cout<<"中点: ";**

**midpoint(p1, p2).print();**

**return 0;**

**}**

**正确答案：   
第1空:**

**Point::Point(double  x1, double  y1)**

**第2空:**

**point.x**

**第3空:**

**point.y**

**第4空:**

**"("<<x<<", "<<y<<")"**

**第5空:**

**sqrt**

**第6空:**

**(p1.x+p2.x)/2 ;     p1.x/2+p2.x/2**

**第7空:**

**(p1.y+p2.y)/2;     p1.y/2+p2.y/2**

**解析：**

**3、三、程序阅读题：**

**1. 阅读以下程序：**

**#include <iostream>**

**using namespace std;**

**class Base1{**

**public:**

**Base1(int d) { cout<<d;}**

**~Base1() {}**

**};**

**class Base2{**

**public:**

**Base2(int d) { cout<<d; }**

**~Base2(){ }**

**};**

**class Derived : public Base1,Base2**

**{**

**public:**

**Derived(int a,int b, int c, int d):Base1(b), Base2(a), b1(d), b2(c) {}**

**private:**

**int b1;**

**int b2;**

**};**

**int main(){**

**Derived d(9,7,8,3);**

**return 0;**

**}**

**输出的结果为：**

**正确答案：   
第1空:**

**79**

**解析：**

**4、三、程序阅读题：**

**2.****阅读以下程序：**

**#include <iostream>**

**using namespace std;**

**class Base**

**{**

**public:**

**void fun1() { cout<<"Base";}**

**virtual void fun2() { cout<<"Base";}**

**};**

**class Derived : public Base {**

**public:**

**void fun1(){ cout<< "Derived" ; }**

**void fun2() { cout<< "Derived"; }**

**};**

**void f(Base& b) { b.fun1(); b.fun2(); }**

**int main(){**

**Derived obj;**

**f(obj);**

**return 0;**

**}**

**输出的结果为：**

**正确答案：   
第1空:**

**BaseDerived**

**解析：**

**5、三、程序阅读题：**

**3.  阅读以下程序：**

**#include<iostream>**

**using namespace std;**

**template<class T>**

**T maxV(T m1,T m2)**

**{return ((m1>m2)?m1:m2);}**

**int main() {**

**cout<<maxV(1,7)<<max(2.0, 5.0);**

**cout<<maxV('y','b')<<endl;**

**return 0;**

**}**

**程序执行后的输出结果为**

**正确答案：   
第1空:**

**75y**

**解析：**

**6、三、程序阅读题：**

**4.  阅读以下程序：**

**#include <iostream>**

**using namespace std;**

**class B**

**{**

**public:**

**virtual void show(){cout<<"B"; }**

**};**

**class D: public B**

**{**

**public:**

**void show(){ cout<<"D";}**

**};**

**void fun1(B \*ptr) { ptr->show(); }**

**void fun2(B &ref) {ref.show(); }**

**void fun3(B b) {b.show(); }**

**int main()**

**{**

**B b,\*p=new D();**

**D d;**

**fun1(p);**

**fun2(b);**

**fun3(d);**

**delete p;**

**return 0;**

**}**

**程序执行后的输出结果为**

**正确答案：   
第1空:**

**DBB**

**解析：**

###### ****三、 简答题（共2题，30分）****

**1、四、分析设计题**

**1.  小林同学想将一个double数据与Complex复数类数据相加，请你设计一个方案或者程序帮她实现。**

**要求：(1)用文字描述你要解决问题的方案，方案不唯一；( 7分）**

**(2) 选择其中一种用代码实现。（8分）**

**正确答案：**

**（1）       文字略**

**（2）       代码实现（答案不唯一）**

**#include<iostream>**

**using namespace std;**

**class Complex**

**{**

**public:**

**Complex() { real =0;imag =0;}**

**Complex( double r, double i) {real =r; imag=i;}**

**operator double( ) {return real;}**

**private:**

**double real;**

**double imag;**

**};**

**int main()**

**{**

**Complex cl(3,4) ,c2(5,-10),c3; //建立 3个Complex 类对象**

**double d;**

**d=2.5+cl;//要求将一个double数据与Complx类数据相加**

**cout<<d<< endl;**

**return 0;**

**}**

**解析：**

**2、四、分析设计题**

**2. 某医院档案室的陈阅红将8个0~100之间的随机数放在数组中，作为开启重要设备的密钥，以二进制的形式写入文件number.txt中，然后因某事故紧急调离，现在急需这些数据开启设备。请你帮忙模拟陈阅红以二进制的形式写入文件，再读出数据显示屏幕上。**

**要求：(1)用文字描述你打算如何处理；( 7分）**

**(2) 用代码实现。（8分）**

**正确答案：**

**参考答案：**

**（1）       文字略**

**（2）       代码实现**

**#include <iostream>**

**#include <fstream>**

**#include<cstdlib>**

**using namespace std;**

**int main(){**

**int arr1[8] , arr2[8];**

**for(int i=0;i<8;i++)**

**{**

**arr1[i]=rand()%101;**

**}**

**ofstream ofile("d:\\number.txt", ios::out | ios::binary);**

**if(!ofile.fail()){**

**ofile.write((char \*)arr1, sizeof(arr1));**

**}**

**ofile.close();**

**ifstream ifile("d:\\number.txt", ios::in | ios::binary);**

**if(!ifile.fail()){**

**ifile.read((char \*)arr2, sizeof(arr2));**

**for(int i=0; i<sizeof(arr2)/sizeof(int); ++i)**

**cout<<arr2[i]<<'\t';**

**cout<<endl;**

**}**

**ifile.close();**

**return 0;**

**}**

**解析：**