79面向对象程序设计\_固定试卷模拟3

**一、单选题 （共20题,共40分）**

1. 当派生类从基类私有继承时，基类中的公有成员和保护成员成为派生类的（2分）

A.保护成员和私有成员 B.公有成员 C.私有成员 D.保护成员

2. 下列符号中能够作为C++标识符的是 （A）

A.points B.2b C.const D.-256

3. 下列关于友元的描述中，错误的是 （2分）

A.如果函数fun()是类A的友元，那么在fun()中可以访问类A的私有成员

B.如果类A是类B的友元，那么类B也是类A的友元

C.如果类A是类B的友元，那么类A的所有成员函数都是类B的友元

D.友元必须在类体内声明

4. C++中，实现封装性需借助于 （2分）

A.枚举 B.类 C.数组 D.函数

5. 若有函数调用fun(x+y, 3, min(a,b))，则fun的实参个数为 （2分）

A.3 B.4 C.5 D.6

6. 下列关于类和对象的描述中，错误的是 （2分）

A.类具有封装性 B. 类对象可以访问类的全部成员

C. 类是所有对象的共同的行为和不同的状态的集合体 D.对象是类的具体实例

7. 下列选项中，不是类成员函数的是 （2分）

A.构造函数 B.析构函数 C.虚函数 D.友元函数

8. 下列程序段中，横线处应填入的内容是 （2分）

class MyClass{

public:

MyClass() { count++; }

private:

static int count;

};

a　　　　　　count=10;

A.int B.static int C.static MyClass::int D.int MyClass::

9. 在C++中，要实现动态联编，调用虚函数时必须使用 （2分）

A.派生类指针 B.基类指针 C.类 D.对象

10. 下列函数中，具有隐含的this指针的是 （B）

void fun1(); //①

class MyClass{

public:

friend void fun2(); //②

static int fun3(); //③

int fun1(); //④

};

A.① B.② C.③ D.④

11. 有如下类定义，obj是类D的对象，下列语句中不违反访问控制权限的是 （2分）

class B{

public: void fun1();

private: void fun2();

protected: void fun3();

};

class D : public B {

protected: void fun4();

};

A.obj.fun1(); B.obj.fun2(); C.obj.fun3(); D.obj.fun4();

12. 下列运算符中，不能被重载的是 2分）

A.\* B.!= C.:: D.++

13. 下列关于常类型的描述中，错误的是 （2分）

A.不能修改常对象的状态 B.通过常对象可以调用一般成员函数

C.常成员函数中不能修改对象的状态 D.必须使用成员初始化列表初始化常数据成员

14. 下列选项中，istream类对象是 （2分）

A.cin B.cout C.cerr D.clog

15. 可以解决程序中代码量不大，但却被频繁调用的函数的调用效率问题的是 （2分）

A.友元函数 B.重载函数 C.虚函数 D.内联函数

16. 下列格式控制符中，可以设置输出项域宽的是 （2分）

A.setw B.setfill C.setprecision D.endl

17.下列对常类型的定义中，错误的是（2分）

A.MyClass const obj(1,2); B.char const \*ptr="hello";

C.const int &ref; D.void fun() const;

18. 已知函数f的原型是void f(int x, int \*y);，变量v1、v2的定义是：int v1, v2;，下列调用语句中，正确的是（2分）

A.f(v1,&v2); B.f(v1,v2); C.f(&v1,v2); D.f(&v1,&v2);

19.下列关于继承和派生的描述中，错误的是 （2分）

A.派生类要向基类的构造函数传递参数

B.多继承时可能会产生二义性

C.派生类可以访问基类的所有数据成员，也能调用基类的所有成员函数

D.一个基类可以有多个派生类，一个派生类可以有多个基类

20. 在下列函数原型中，可以作为类A构造函数的是 （2分）

A.void A(int); B.int A(); C.A(int) const; D.A(int);

**二、填空题 （共10题,共20分）**

1. 在C++中，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_是指同一个函数名可以对应多个函数的实现。 （2分）

2. 下列语句序列执行后输出10，请将划线处的语句补充完整。 （2分）

class MyClass {

public:

MyClass(int x):val(x) { }

void Print() { cout<<val<<endl; }

private :

int val ;

};

int main() {

MyClass obj(10);

　　　　Print();

return 0;

}

3. 已知一个函数的原型是double fun(double x);，若要以4.25为实参调用该函数，应使用表达式\_\_\_\_\_\_\_\_\_\_\_\_ 。 （2分）

4. 下列语句序列执行后输出Hello，请将划线处的语句补充完整。 （2分）

class MyClass {

public:

void Print() const { cout<<"Hello"; }

};

int main() {

MyClass\* p = new MyClass();

　　　　Print();

return 0;

}

5.有两个类M和C，其中类C定义如下：class C { public: M m;} ;。若建立类C的对象object，则对象m和对象object中先被初始化的是对象\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。 （2分）

6. 请在划线处填写构造函数的声明，使得语句Point p1;创建的p1对象为原点。 （2分）

class Point {

public:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

private:

int x,y;

};

7. 请在划线处写出MyClass类的析构函数声明。 （2分）

class MyClass {

public:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

};

8. 设置虚基类的目的是为了解决多继承产生的\_\_\_\_\_\_\_\_\_\_\_\_问题。 （2分）

9. 将一个函数声明为一个类的友元函数必须使用关键字\_\_\_\_\_\_\_\_\_\_\_\_\_\_（2分）

10. 派生新类的类称为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，而派生出的新类称\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**三、改错题 （共1题,共10分）**

下列程序中有三个错误，请改正错误（注意不要修改主函数），使程序的输出结果为：

Constructor

The value is 10

Destructor

源文件清单如下：

Line1: #include <iostream.h>

Line2: class MyClass

Line3: {

Line4: public:

Line5: MyClass(int x):value(x) { cout<<"Constructor"<<endl; }

Line6: void ~MyClass() { cout<<"Destructor"<<endl; }

Line7: void Print() const;

Line8: private:

Line9: int value=0;

Line10: };

Line11: void MyClass::Print()

Line12 {

Line13 cout<<"The value is "<<value<<endl;

Line14: }

Line15: int main()

Line16: {

Line17: const MyClass object(10);

Line18: object.Print();

Line19: return 0;

Line20: }

**四、阅读程序题 （共4题,共30分）**

**1. 请写出下列程序的输出结果。**

#include <iostream.h>

class Base {

public:

virtual void f() { cout<<"fB"<<endl; }

virtual void g() { cout<<"gB"<<endl; }

};

class Derived : public Base {

public:

virtual void f() { cout<<"fD"<<endl; }

virtual void g() { cout<<"gD"<<endl; }

};

int main() {

Base b,\*p=new Derived;

b.f();

p->g();

return 0;

}

2. 请写出下列程序的输出结果。 （10分）

#include <iostream.h>

class Base1 {

public:

Base1() { cout<<"Base1"<<endl; }

~Base1() { cout<<"~Base1"<<endl; }

};

class Base2 {

public:

Base2() { cout<<"Base2"<<endl; }

~Base2() { cout<<"~Base2"<<endl; }

};

class Derived : public Base1, public Base2 {

public:

Derived() { cout<<"Derived "<<endl; }

~Derived() { cout<<"~Derived"<<endl; }

private:

Base1 b1;

Base2 b2;

};

int main() {

Derived d;

return 0;

}

3. 请写出下列程序的输出结果。 （6分）

#include <iostream.h>

class A {

public:

A(int i):r1(i) { cout<<r1<<endl; }

~A()　　　　 { cout<<'~'<<r1<<endl; }

void print() {cout<<"Empty:"<<r1<<endl;}

void print() const {cout<<"Const:"<<r1<<endl;}

void print(int x) {cout<<"Param:"<<x\*x<<endl;}

private:

int r1;

};

int main() {

A a1(1);

const A a2(2);

a1.print();

a2.print();

return 0;

}

4. 请写出下列程序的输出结果。 （6分）

#include <iostream.h>

class Point{

public:

Point() { number++; cout<<number<<endl; }

~Point() { number--; }

static int GetNumber() { return number; }

private:

static int number;

};

int Point::number=0;

int main() {

Point A,B;

Point \*ptr=new Point[3];

delete[] ptr;

cout<<Point::GetNumber()<<endl;

return 0;

}