

C++期末考试试卷

2019 -- 2020 学年 2 学期

时间 100 分钟

2020 年 7 月 11 日

面向 对象编程(C++) 课程 48 学时 3 学分 考试形式: 闭 卷

专业年级: 2019 级自动化与电气类 总分 100 分

注: 此页不作答题纸, 请将答案写在答题纸上

一、填空题(每空 1 分, 共 13 分)

- 假定A为一个类, 则执行语句: A a(3), b[4], *p[5];时, 调用该类构造函数 (1) 次。
- 假定Bd为一个类, 请写出Bd的拷贝构造函数的声明语句的一种形式: (2) 。
- 含有纯虚函数的类称为 (3) 。
- 不同对象可以调用相同名称的函数, 但执行完全不同行为的现象称为 (4) 。
- 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。则运算符重载函数operator+的参数个数为 (5) , operator*的参数个数为 (6) 。
- 要在某个类中定义一个名字为vir_fun 的纯虚函数, 该函数没有参数, 返回类型为int, 则该纯虚函数的说明形式为: (7) 。
- 假定一个类A有两个int型的数据成员a和b, 且类A有如下两个构造函数:
A(int aa, int bb, int cc = 3) {a = ++aa; b = a * bb + cc;}
A(char aa, char bb) { a = aa - 'a'; b = a * (bb - 'a');}
则创建对象A x1(4, 5), x2('e', 'f');后, x1.b的值为 (8) ; x2.b的值为 (9) 。
- 若派生类的成员函数不能直接访问基类中继承来的某个成员, 则从访问权限来看, 该成员一定是基类中的 (10) 成员。
- 如果要把类B的成员函数void fun()说明为类A的友元函数, 则应在类A中加入语句 (11) 。
- 已知有一个int类型的指针变量p, 使用new为动态分配10个存储空间给p的语句为: (12) 。
- 在删除一个动态对象时, 将自动调用该动态对象所属类的 (13) 函数。

二、程序阅读题(每小题 5 分, 共 15 分)

1. 阅读下列程序写出运行结果

```
#include<iostream>
using namespace std;
class A{
public:
    A(){x=0;y=0;}
    A(int i, int j) {x=i;y=j; }
    void copy(A *a);
    void setxy (int i, int j) {
        x=i; y=j;
    }
    void print(){
        cout<<x<<" "<<y<<endl;
    }
private:
    int x,y;
}; // (右侧后续)
```

```
void A::copy(A *a){
    x=a->x;
    y=a->y; }
void fun(A a1, A &a2)
{
    a1.setxy(12,15);
    a2.setxy(22,25);
}
int main()
{
    A p(5,7), q;
    q.copy(&p);
    q.print();
    fun(p, q);
    p.print();
    q.print();
    return 0; }
```

2. 阅读下列程序写出运行结果

```
using namespace std;
class A
{
    int number, price, total;
public:
    A(int n, int p = 20)
    {
        number = n;
        price = p;
    }
    A()
    {
        number = 20;
        price = 10;
    }
    void print_result()
    {
        cout << "number = " << number;
        cout << ", price = " << price;
        cout << ", total = " << number*price << endl;
    }
};
main()
{
    A op[5]={A(), A(20), A(10, 12), A(15), A()};
    for(int i=0;i<5;i++)
    {
        op[i].print_result();
    }
}
```

3. 阅读下列程序写出运行结果

```
#include<iostream>
using namespace std;
class A
{
    int x;
public:
    A(int x){ A::x=x;
        cout<<"class A"<<endl; }
};
class B
{
    A a1;
public:
    B(int x):a1(x)
    { cout<<"class B"<<endl; }
};
class C:public B
{
    A a2;
public:
    C(int x):B(x),a2(x)
    { cout<<"class C"<<endl; }
};
class D:public C
{
public:
    D(int x):C(x)
    { cout<<"class D"<<endl; }
};
main()
{ D dobj(10); }
```

三、程序补全题：下列程序缺少若干语句，在空白处填上合适的代码（每空 2 分，共 20 分）

1. 下面的程序中，定义了一个时间类 Time，Time 中包含三个 int 的数，分别表示时钟、分钟、秒钟。Time 类中定义了两个构造函数，其中普通的构造函数初时间为 1 点 1 分 1 秒其中的拷贝构造函数函数将一个时间复制给另一个时间，另外定义一个友元函数求两个时间之间差了多少秒。

```
#include<iostream>
#include<math.h>
using namespace std;
class Time
{
    int hour, minute, second;
public:
    _____ (1) _____
    { hour = h; minute = m; second = s; }
    Time(Time &t)
    { _____ (2) _____ }
    _____ (3) _____
};
int operator-(Time &T1, Time &T2)
{
    int s = T1.second - T2.second;
    s += 60*(T1.minute - T2.minute);
    s += 3600*(T1.hour - T2.hour);
    return(abs(s));
}
int main()
{
    Time T1(2,3,4), T2(3,4,5), T3(T2);
    cout << "time span is " << T1 - T3 << " second";
    return 0; }
}
```

2. 定义一个类表示 3*3 的矩阵，在类中对乘法运算符进行重载，求两个 3*3 矩阵的乘积。

（设矩阵 A[3][3]，B[3][3]，其乘积 C = A*B 仍为 3*3 的矩阵， $c_{ij} = \sum_{k=1}^3 \sum_{k=1}^3 (a_{ik}b_{kj})$ ）

```
#include<iostream>
using namespace std;
class matrix
{
    int a[3][3], i, j, k;
public:
    matrix(int b[3][3]) {
        for(i = 0; i < 3; i++)
            for(j = 0; j < 3; j++) a[i][j] = b[i][j]; }
    matrix() {
        for(i = 0; i < 3; i++)
            for(j = 0; j < 3; j++) a[i][j] = 0; }
    matrix operator*(matrix &M1) {
        matrix M = matrix();
        for(int i = 0; i < 3; i++)
            for(j = 0; j < 3; j++) {
                _____ (4) _____
                for(k = 0; k < 3; k++) _____ (5) _____
            }
        _____ (6) _____
    }
};
int main()
{
    int A[3][3] = {1,2,3,4,5,6,7,8,9};
    int B[3][3] = {1,0,0,0,1,0,0,0,1};
    matrix MA(A),MB(B),MC=MA*MB;
    return 0; }
}
```

<p>3. 在空白处补上合适代码，使输出结果为：3，4，5。</p> <pre> #include<iostream> #include <math.h> using namespace std; class test { double x; public: test(double d=0.00):x(fabs(d)) { } _____ (7) _____ { return test(sqrt(x*x+c.x*c.x)); } _____ (8) _____ { stream<<c.x; return stream; } }; int main() { test ma1(3), ma2(-4), ma3 = ma1 + ma2; cout << ma1 << ", " << ma2 << ", " << ma3 << endl; return 0; } </pre>	<p>4. 定义虚函数，利用基类指针访问派生类同名函数:</p> <pre> #include<iostream> using namespace std; class Shape { public: _____ (9) _____ void sname() {cout<<"Shape";} }; class Circle: public Shape { public: char type; Circle (char c) { type = c;} void sname() { cout << type;} }; int main() { Circle ns('p'); _____ (10) _____ Sp = &ns; Sp->sname(); return 0; } </pre>
---	--

四、简答题（每小题 5 分，共 15 分）

1. 什么是内联函数？内联函数有些什么特点？
2. 面向对象编程中有哪三种继承方式，请说明它们之间的区别。
3. 什么是抽象类？请说明抽象类的作用？

五、程序改错题，找到错误行，分析原因并改正(每个错误 2 分，共 14 分)

<p>1、本小题2处错误</p> <pre> #include <iostream> using namespace std; (1) class A (2) { (3) int x; (4) double y; (5) public: (6) A() {x = 0, y = 0;} (7) A(int i=1, double j=1.5) { x = i; y = j; } (8) void set_x(int i){x=i;} (9) }; (10) int main() (11) { A a, b(1, 0.5); (12) a.set_x(2); (13) a.y=1.8; (14) return 0; (15) } </pre>	<p>2、本小题2处错误</p> <pre> #include <iostream> using namespace std; (1) class X (2) { (3) int a, b; (4) public: (5) void setX(int i, int j){a=i; b=j;} (6) friend void display(X &); (7) }; (8) void display(X& x) (9) {cout<<this->a<<","<<this->b<<endl;} (10) int main() (11) { X a; (12) a.setX(1,2); (13) a.display(); (14) return 0; (15) } </pre>
---	--

<p>3.对本小题的3个地方进行修改使程序能正确运行并得到结果为：</p> <pre> Base display a=1 Derived display 1) #include<iostream> 2) using namespace std; 3) class Base{ 4) int a; 5) public: 6) Base(int c=1){ a=c; } 7) void display() 8) { 9) cout<<"Base display"<<endl; 10) cout<<"a="<<a<<endl; 11) } 12) }; //（右侧后续） </pre>	<pre> 13) class Derived: public Base 14) { 15) public: 16) void display(){ 17) cout<<"Derived display"<<endl; 18) cout<<"a="<<a<<endl; 19) } 20) }; 21) 22) void display (Derived & rr) 23) { 24) rr.display(); 25) } 26) 27) int main() 28) { 29) Base b; 30) Derived d; 31) display (b); 32) display(d); 33) return 0; 34) } </pre>
--	---

六、 综合题(第 1 题 15 分，第二题 8 分，共 23 分)

1、对不同类型的动物抽象，定义抽象基类 `animal`（数据成员包括动物名字，颜色；成员函数自己考虑），由它派生出两个派生类：鸟类（数据成员姓名、颜色和飞行速度，及函数成员）、鱼类（数据成员姓名、颜色和游泳速度），由鸟类动物和鱼类动物派生出水鸟类动物；编写构造函数对成员进行初始化；定义虚函数可以分别对不同类动物的速度进行报告（屏幕输出）。编写主函数对上述功能进行测试，实现动态多态性。（要求数据成员需定义为私有或保护类型）

2. 某商店经销一种货物，货物成箱购进，成箱卖出，购进和卖出时以重量为单位，不同箱子重量不一样。因此，商店需要记录库存货物的总重量，现在编写程序对商店货物购进和卖出的情况进行管理。通过常数据成员记录货物的总箱数和总重量，并对当前货物信息进行显示。基本程序如下，按要求进行补充和修改：

- 1) 根据需求，补充静态数据成员并对其进行初始化。
- 2) 如果有进货或卖出则实时更新总箱数和总重量
- 3) 对所有成员函数，考虑哪一个可以修改避免其误改变数据成员的值，并进行修改。

```

(1) #include<iostream>
(2) using namespace std;
(3) class Goods
(4) {
(5)     public:
(6)     Goods(int inBox, double inWeight);
(7)     ~Goods();
(8)     void Sell(int outBox, double outWeight);
(9)     void print();
(10)    private:
(11)    double Weight;

```

```

(12) int BoxNum;
(13) _____
(14) _____
(15) };

(16) Goods::Goods(int inBox, double inWeight) //进货
(17) {
(18) BoxNum = inBox;
(19) Weight = inWeight;
(20) _____
(21) _____
(22) }

(23) void Goods::Sell(int outBox, double outWeight) //卖出
(24) {
(25) _____
(26) _____
(27) }

(28) void Goods::print() {
(29) cout << "当前货物总箱数为: " << totalBox << "箱" << endl;
(30) cout << "当前货物总重量为: " << totalWeight << "kg" << endl;
(31) }
(32) Goods::~~Goods()
(33) {
(34) }
(35) _____
(36) _____

(37) int main()
(38) {
(39) Goods gd(10, 300);
(40) gd.Sell(3, 60);
(41) gd.print();
(42) return 0;
(43) }

```