

姓名: _____

学号: _____

部(院): _____

_____ 级 _____ 班

大 连 理 工 大 学

课程名称: 面向对象方法与 C++ 程序设计 试卷: A 考试形式: 闭 卷

授课部(院): 软件学院 考试日期: 2020 年 8 月 18 日 试卷共 11 页

	一	二	三	四	五	总分
标准分	30	20	15	20	15	100
得 分						

得 分	
--------	--

一、单项选择题 (30 分, 15 小题, 每题 2 分)

- 以下哪组函数能实现合法的函数重载 ()。
【A】void fun(); int fun(); 【B】void fun(int* p); void fun(int& r);
【C】double fun(double*); void fun(double []); 【D】char fun(char); char fun(char&);
- 关于类与对象的理解, 正确的描述是 ()。
【A】一个类只能有一个对象, 一个对象只能属于一个类
【B】同类对象具有相同的数据成员, 可以调用相同的成员函数
【C】一个类可以实例化多个对象, 一个对象可以属于多个类
【D】具有相同的成员数据与成员函数的对象, 必定属于同一个类
- 关于引用, 下列说法不正确的是 ()。
【A】引用相当于为其所引用的变量起别名 【B】引用通常用于函数传参和函数返回值
【C】引用占用独立的内存空间 【D】引用在创建时必须初始化
- 在类 A 中实现+运算符重载, 以下描述正确的是 ()。
【A】形如 A=A+2 的加法运算, 不能通过成员函数实现重载
【B】形如 A=2+A 的加法运算, 不能通过成员函数实现重载
【C】形如 A=A+2 的加法运算, 不能通过友元函数实现重载
【D】形如 A=2+A 的加法运算, 不能通过友元函数实现重载
- 下列描述错误的是 ()。
【A】类中的成员函数可以进行重载 【B】inline 内联函数可以提高程序效率
【C】类的成员函数可以设置默认参数 【D】类的成员函数都具有 this 指针
- 下列关于类的继承描述中, 正确的是 ()。
【A】公有继承情况下, 派生类可以访问其基类的所有成员
【B】可以将派生类指针指向一个基类对象
【C】派生类对象中包含基类的所有数据成员
【D】一个派生类只能有一个基类

7. 有如下类 A，以下描述正确的是（ ）。

```
class A{
    const int data;
public:
    A( int );
    void fun1( )const;
    void fun2( const int & );
};
```

- 【A】 void A::fun2(const int& r){ r = data; } 【B】 A::fun1()可以调用 A::fun2();
【C】 A 类中可以增添 void fun1();实现函数重载 【D】 A::A(int d){data = d;}
8. 已知类 A 具有构造函数 A::A(int=0)，不能创建一个 A 类的对象的语句是（ ）。
【A】 A a; 【B】 A a();
【C】 A* a = new A; 【D】 A* a = new A(1);
9. 有如下类 A，对友元函数正确的描述是（ ）。

```
class A{
    int data;
public:
    .....
    friend void fun( A&, int );
};
```

- 【A】 void fun(A& a, int d){ a.data = d; } 【B】 A a1, a2; a1.fun(a2,3);
【C】 void A::fun(A& a, int d){ a.data = d; } 【D】 fun()的访问权限为 public
10. 下列语法不支持多态的为（ ）。
【A】 虚函数 【B】 虚继承 【C】 函数重载 【D】 运算符重载
11. 关于类中的静态成员，以下描述正确的是（ ）。
【A】 静态成员函数可通过 this 指针访问静态成员数据
【B】 静态成员函数可以访问类中非静态成员数据
【C】 静态成员数据只能在构造函数中初始化
【D】 静态成员数据可以在非静态成员函数中被访问
12. C++实现动态多态，不是必要条件的为（ ）。
【A】 公有继承体系
【B】 基类声明虚函数，派生类定义虚函数
【C】 派生类对象调用虚函数
【D】 基类指针或者引用对象调用虚函数
13. 关于类的封装性，以下理解正确的为（ ）
【A】 类中封装了数据与函数
【B】 封装可以实现私有信息的隐藏与保护
【C】 封装是实现继承的基础
【D】 以上描述都正确

14. 对于如下类，描述正确的是（ ）。

- 【A】 Plant 为虚基类，它是 Tree 类的公有基类
- 【B】 Plant 为抽象类，不能实例化 Plant 对象
- 【C】 Tree 为抽象类，所有成员都是纯虚函数
- 【D】 Tree 不包含纯虚函数，能实例化该类对象

<pre>class Plant{ protected: int age; public: virtual void sprout() = 0; virtual void leaf() = 0; virtual void flower() = 0; };</pre>	<pre>class Tree : public Plant { float height; public: void sprout() {}; void leaf() {}; void fruit() {}; };</pre>
---	--

15. 下列函数模板的声明正确的是（ ）。

- 【A】 template <class T1, T2 > T2 fun(T1 t1);
- 【B】 template <typename T >T fun(int a);
- 【C】 template <class T1, class T2> void fun(T1 t1, T2 t2);
- 【D】 template < typename T1, T2> void fun(T1 t1, T2 t2) ;

得分

二、程序阅读（20 分，共 4 题）

写出下面程序的运行结果

1. （4 分）

```
#include<iostream>
using namespace std;
class Plant {
protected:
    char name[20];
public:
    Plant(const char* n = "no name") {
        if(n!=NULL) strcpy(name, n);
        cout << "construct Plant " << name << endl;
    }
    ~Plant() {
        cout << "destruct Plant " << name << endl;    }
    void show() {
        cout << "Plant is " << name << endl;    }
};
int main() {
    Plant *p1 = new Plant("cherry tree");
    Plant p2("apple tree");
    Plant *p3 = &p2;
    p3->show();
}
```

2. (5 分)

```
#include<iostream>
using namespace std;

class Flower {
private:
    char color[20];
public:
    Flower(const char* c = "no color") {
        if(c!=NULL) strcpy(color, c);
        cout << "construct Flower " << color << endl;
    }
    void setColor(const char* c = "no color") {
        if(c!=NULL) strcpy(color, c);
        cout << "set color " << color << endl;
    }
    ~Flower() {
        cout << "destruct Flower " << color << endl;
    }
};

class FruitTree {
protected:
    char fruit[20];
    Flower f;
public:
    FruitTree(const char* fru, const char* c){
        strcpy(fruit, fru);
        f.setColor(c);
        cout << "construct FruitTree " << fruit << endl;
    }
    ~FruitTree() {
        cout << "destruct FruitTree " << fruit << endl;
    }
};

int main() {
    FruitTree ft("cherry", "pink");
}
```

3. (6分)

```
#include<iostream>
using namespace std;

class Plant {
protected:
    char name[20];
public:
    Plant(const char* n = "no name") {
        strcpy(name, n);
        cout << "construct Plant " << name << endl;
    }
    ~Plant() {
        cout << "destruct Plant " << name << endl;
    }
};

class FruitTree : public Plant {
protected:
    char fruit[20];
public:
    FruitTree(const char* n = "no name", const char* fru = "no name") {
        strcpy(name, n);
        strcpy(fruit, fru);
        cout << "construct FruitTree " << fruit << endl;
    }
    void disp() {
        cout << "name: " << name << endl;
        cout << "fruit: " << fruit << endl;
    }
    ~FruitTree() {
        cout << "destruct FruitTree " << fruit << endl;
    }
};

int main() {
    FruitTree ft("cherry tree", "cherry");
    ft.disp();
}
```

4. (5 分)

```
#include<iostream>
using namespace std;
const float PI = 3.14f;

class MyException {
public:
    string message;
    MyException(string msg) :message(msg) {    }
};

float Div(int m, int n) {
    if (n == 0)
        throw MyException("Devided by 0.");
    return (float) m / n;
}

float Area(float r) {
    try {
        if (r < 0)
            throw(-1);
        cout << "Area calculated." << endl;
        return PI * r * r;
    }
    catch (int) {
        cout << "Int exception caught!" << endl;
        return -1;
    }
}

int main() {
    float r = -1.0;
    try {
        cout << "Start try Block" << endl;
        Area(r);
        cout << "End of Area Block" << endl;
        Div(2, 0);
        cout << "End of Try Block" << endl;
    }
    catch (MyException& e) {
        cout << "MyException caught!" << endl;
        cout << e.message << endl;
    }
    return 0;
}
```

得分	
----	--

三、程序分析（15 分, 共 3 题）

分析程序, 回答问题

1. (5 分)。设计 Paper 类, 在横线处给出相关代码。

```
#include<iostream>
using namespace std;
class Paper {
    static int count;    //Paper数量
    int pages;          //Paper页数
public:
    Paper(int pages);
    ~Paper() { count --; }
    static int getCount();
    void print() const;
};
_____ ① _____ //初始化count=0
Paper::Paper(int pages)
{
    count++;
    _____ ② _____ //Paper页数赋值
}
_____ ③ _____ //类外实现成员函数getCount
{
    return count;
}
_____ ④ _____ //类外实现成员函数print
{    cout << pages << endl;    }
int main() {
    cout << "The number of papers is " << _____ ⑤ _____ << endl; //输出 Papers 的数量
    Paper p(10);
    p.print();
}
```

2. (5 分) 分析以下代码, 找出其中的语法错误或逻辑错误并改正。

回答的格式为: 行号: 正确的写法。

改正后, 程序输出结果为:

A Circle with radius 1

The area is : 3.14

A square with side 1

The area is : 1

```

1) #include<iostream>
2) using namespace std;
3) const float PI = 3.14f;
4) class Point {
5)     float x, y;
6) public:
7)     Point(float x = 0, float y = 0):x(x), y(y) { }
8) };
9) class Shape {
10)     Point center;
11) public:
12)     Shape(float x = 0, float y = 0):center(x, y){ }
13)     float area() { return 0; };
14)     void print() = 0;
15) };
16) class Circle : public Shape{
17) private:
18)     float radius;
19) public:
20)     Circle(float x=0, float y=0, float r = 1.0f):Shape(x, y) {
21)         radius = r > 0 ? r : 0;
22)     }
23)     float area() { return PI * radius* radius; }
24)     void print() {
25)         cout << "A Circle with radius " << radius << endl;
26)     }
27) };
28) class Square : public Shape {
29)     float side;
30) public:
31)     Square(float x, float y, float a = 1.0f): Shape(x, y) {
32)         side = a > 0 ? a : 0;
33)     }
34)     float area() { return side * side; }
35)     void print() {
36)         cout << "A square with side " << side << endl;
37)     }
38) };
39) int main(){
40)     Shape ptr[2] = { new Circle(), Square(4,1) };
41)     for (int i = 0; i < 2; i++) {
42)         ptr[i].print();
43)         cout << "The area is : " << ptr[i]->area() << endl; }
44) }

```


3. (5 分) 分析代码, 回答以下问题。

```
#include<iostream>
using namespace std;
class Plant{
protected:
    int age;
};
class Tree : public Plant{
    float height;
public:
    void setAge(Plant &p) {
        age = p.age;    // ①
    }
    void setHeight(Tree &t) {
        height = t.height; // ②
    }
    friend void test();
};
void test() {
    Tree tr;
    tr.age = 3;    // ③
    tr.height = 3.2f; // ④
}
```

- (1) Tree 类中包含哪些数据成员? 他们的访问权限分别是什么?
- (2) 代码中标注的①②③④四条语句中, 不正确的有哪些?

得	
分	

四、编程题 (20 分, 共 3 题)

1. (6 分) 1. 设计一个函数模板 $T^* \text{Merge}(T^*a, \text{int } n, T^*b, \text{int } m)$; 合并两个增序数组 (数组长度分别为 n 和 m) 成为一个增序数组, 合并后增序数组长度为 $n+m$, 如有重复数据则全部保留。在 `main` 函数中使用数组 `int a[] = {1,5,7,9}`, `int b[] = {2,4,6,7,8}` 测试该模板, 输出结果为 1,2,4,5,6,7,7,8,9。请给出完整的函数模板的定义。

2. (8 分) 设计一个处理器类 (CPU), 包含私有成员品牌 (`_brand`)。构造 CPU 对象时要求输入品牌信息。设计一个电脑类 (Computer), 包含私有成员处理器 (`_cpu`) 和尺寸 (`_size`)。构造电脑类对象时要求输入处理器品牌和电脑尺寸。请在类外实现 CPU 类中的成员函数 (构造函数和 `show` 函数), 并写出 Computer 类的定义与实现 (类内类外均可), 使 `main` 函数可以运行得到相应的结果。

```
#include<iostream>
using namespace std;
```

```

class CPU {
    char* _brand;
public:
    CPU(const char* b);
    void show();
    ~CPU(){ if(_brand) delete []_brand; }
};
// CPU 类成员函数的实现
.....
// Computer 类的定义与实现
class Computer {
    .....
};
int main(){
    Computer com("Intel", 15);
    com.show(); //输出: CPU 品牌为 Intel;电脑尺寸为 15
}

```

3. (6 分) 设计如下箱子 Bin 类，并写出该类成员函数的声明与定义（类内类外均可），使程序运行得到相应的结果。

```

#include <iostream>
using namespace std;
class Bin {
    int length; //长
    int width;   //宽
    int height;  //高
public:
    Bin(int x = 0, int y = 0, int z = 0)
    {
        length = x>0? x:0;
        width = y>0? y:0;
        height = z>0? z:0;
    }
    ..... //成员函数的声明
};
// Bin 类成员函数的实现
.....
int main(){
    Bin b1(1, 3, 5), b2(2, 1, 4);
    if (b1 < b2) //比较两个箱子的体积
        cout << b1; //显示箱子的长宽高信息，输出: 1 3 5
    else
        cout << b2; //显示箱子的长宽高信息，输出: 2 1 4
    return 0;
}

```

得分	
----	--

五、面向对象程序设计（15 分，共 1 题）

利用面向对象方法设计一个方舱医院管理程序框架。方舱医院中的人员（Person）有病人（Patient）和医院工作人员（Employee），医院工作人员又分为管理人员（Manager）和医生（Doctor）；每一类人都有姓名 name 与编号 id；每名医生同时负责 8 位病人；病人（Patient）需要记录其病床号与病情状态信息；医生需要保存是否在岗与负责的病人的信息；管理人员需要保存部门与职能等信息。每类人员都要定期报告自己的相关信息。

- 1) 上述描述中可以抽象出哪些类？（5 分）
- 2) 合理设计上述类的关系，分析哪些类之间存在继承关系或组合/聚合关系。（5 分）
- 3) 若有如下方舱医院类 MCHospital,

```
class MCHospital {
    Person* persons[1000]; // 存放各种人员
    int num; // 人员数目，最多 1000 人
public:
    void addPerson(Person& p); // 添加各类人员
    void showInfo() {
        for (int i = 0; i < num; i++)
            persons[i]->report(); // 报告个人信息
    };
};
```

调用 showInfo() 函数可以输出各类人员的信息，请给出人员基类 Person 类的声明，包括本题中涉及到的最主要数据成员和成员函数，注意哪个成员函数需要声明为虚函数。（5 分）