

学号:

学生姓名:

年级:

专业:

院(系):

线

封

密

华中师范大学 2017-2018 学年第一学期

期末考试试卷 (A 卷)

课程名称: 面向对象程序设计 课程编号: 48740008 任课教师: _____

题型	选择题	简答题	程序分析	编程题	总分
分值	30	25	15	30	100
得分					

得分	评阅人

一、选择题。(共 15 题, 每小题 2 分, 共 30 分)

- () 1. 下面说法哪种是对的?
- A. 常引用所引用的变量, 其值不能被修改 B. 不能通过常量指针, 去修改其指向的变量
- C. 常量指针一旦指向某个变量, 就不能再指向其他变量 D. 以上都不对
- () 2. 下面说法正确的是:
- A. 多个重载函数的参数个数必须不同。 B. 两个函数, 参数表相同, 返回值类型不同, 它们是重载关系。
- C. 调用一个第二个和第三个参数都有缺省的函数时, 可以不写第二个实参而写第三个实参。 D. 使用内联函数的目的是提高程序的运行速度
- () 3. 假设 A 是一个类的名字, 下面的语句生成了几个类 A 的对象?
- A * arr[4] = { new A(), NULL, new A() };
- A. 1 B. 2 C. 3 D. 4
- () 4. 有类 A 如下定义, 下面哪条语句是编译不会出错的?
- ```
class A {
 int v;
 public:
 A (int n) { v = n; }
};
```
- A. A a1(3);    B. A a2;    C. A \* p = new A();    D. 以上均会编译报错
- ( ) 5. 假设 A 是一个类的名字, 下面哪段程序不会调用 A 的拷贝构造函数?
- A. A a1, a2; a1 = a2;    B. void func( A a ) { cout << "good" << endl; }
- C. A func() { A tmp; return tmp; }    D. A a1; A a2(a1);



( ) 9. 若变量已正确定义并具有初值, 下列表达式合法的是

```
int main() {
 int x(1), a(0), b(0);
 switch(x) {
 case 0: b++;
 case 1: a++;
 case 2: a++; b++;
 }
 cout<<"a="<<a<<" ,b="<<b;
 return 0;
}
```

- A. a=2, b=1      B. a=1, b=1      C. a=1, b=0      D. a=2, b=2

( ) 10. 函数原型为 fun(int (\*p) [3],int), 调用形式为 fun(a,2), 则 a 的定义应该为

- A. int \*\*a      B. int (\*a)[]      C. int a[][3]      D. int a[3]

( ) 11. int i=100; 下列引用方法中, 正确的是

- A. int &r=i;      B. int &r=100;      C. int &r;      D. int &r=&i;

( ) 12. 在 C++ 中, 函数默认的存储类别为

- A. auto      B. static      C. extern      D. 无存储类别

( ) 13. 下列程序段的运行结果为

```
#define ADD(x) x+x
int main()
{
 int m=1, n=2, k=3;
 int sum=ADD(m+n)*k;
 cout<<"sum="<<sum;
 return 0;
}
```

- A. sum=9      B. sum=10      C. sum=12      D. sum=18

( ) 14. 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

( ) 15. 设置虚基类的目的是

- A. 简化程序      B. 消除二义性  
C. 提高运行效率      D. 减少目标代码

( ) 16. 如果表达式 ++i\*k 中的“++”和“\*”都是重载的友元运算符, 则采用运算符函数调用格式, 该表达式还可表示为

- A. operator\*(i.operator++(), k)      B. operator\*(operator++(i), k)  
C. i.operator++().operator\*(k)      D. k.operator\*(operator++(i))



( ) 21. 如下程序的运行结果是哪一个?

```
#include <iostream>
using namespace std;
class BASE{
 public:
 ~BASE(){ cout<<"BASE";}
};
class DERIVED: public BASE {
 public:
 ~DERIVED(){ cout<<"DERIVED";}
};
int main(){DERIVED x; return 0 ;}
```

- A. BASE                      B. DERIVED                      C. BASEDERIVED                      D. DERIVEDBASE

( ) 22. 类模板的使用实际上是将类模板实例化成一个

- A. 函数                      B. 对象                      C. 类                      D. 抽象类

( ) 23. 类模板的实例化

- A. 在编译时进行                      B. 属于动态联编  
C. 在运行时进行                      D. 在连接时进行

( ) 24. 以下类模板正确的实例化方式为

```
template <class T1, class T2=int, int num=10>
class Tclass {...};
```

- A. Tclass <char &, char> C1                      B. Tclass <char \*, char, int> C1;  
C. Tclass <> C1;                      D. Tclass <char , 100, int> C1

( ) 25. 定义 char \*p="abcd", 能输出 p 的值("abcd"的地址)的为

- A. cout<<&p;                      B. cout<<p;  
C. cout<<(char \*)p;                      D. cout<< const\_cast<void \*>(p);

( ) 26. 当使用 fstream 定义一个文件流, 并将一个打开文件的文件与之连接, 文件默认的打开方式为

- A. ios::in                      B. ios::out  
C. ios::in|ios::binary                      D. ios::out|ios::binary

( ) 27. 若文件长度为 16 个字节, 执行如下代码的返回值为

```
myfile.seekg(-10, ios::end);
myfile.read((char *)(&c), sizeof(long)); myfile.tellg();
```

- A. 7                      B. 4                      C. 5                      D. 2

( ) 28. char \* S0="12345", 对 string 类串 s1 初始化错误的是

- A. string s1=S0;                      B. string s1(S0);  
C. string s1(S0, 0, 3);                      D. string \*s1=S0;

( ) 29. 下列关于 this 指针的叙述中, 正确的是

- A. 任何与类相关的函数都有 this 指针  
B. 类的成员函数都有 this 指针  
C. 类的友元函数都有 this 指针  
D. 类的非静态成员函数才有 this 指针

( ) 30. 下列程序段的运行结果为

```
#include <iostream>
using namespace std;
class A{
public:
 A(){cout<<"A";}
 ~A(){cout<<"-A";}
};
class B:public A{
 A* p;
public:
 B(){p = new A(); cout<<"B"; }
 ~B(){delete p; cout<<"-B"; }
};
int main(){
 B obj;
 return 0;
}
```

- A. AAB-A-B-A  
C. BAA-B-A-A

- B. ABA-B-A-A  
D. ABA-A-B-A

| 得分 | 评阅人 |
|----|-----|
|    |     |

## 二、简答题。(共 4 题, 每小题 5 分, 共 20 分)

1. 什么是浅拷贝? 什么是深拷贝? 二者有何异同? 试举例并画图说明。

2. C++中静态成员有何作用？它有何特点？静态函数里能否直接调用非静态函数？
3. 在 C++中，能否声明虚构造函数？为什么？能否声明虚析构函数？为什么？如果析构函数没有申明为 `virtual` 会导致什么问题？试举例说明。
4. 什么是迭代器，其作用是什么？

|    |     |
|----|-----|
| 得分 | 评阅人 |
|    |     |

### 三、程序分析题。(共 4 题, 每小题 5 分, 共 20 分)

1. 分析下面的程序, 写出程序运行的结果。

```
#include<iostream>
using namespace std;
class MyClass {
public:
 int number;
 void set(int i);
};
int number=3;
void MyClass::set (int i){
 number=i;
}
void main() {
 MyClass my1;
 int number=10;
 my1.set(5);
 cout<<my1.number<<endl;
 my1.set(number);
 cout<<my1.number<<endl;
 my1.set(::number);
 cout<<my1.number;
}
```

2. 下面函数模板求  $x^n$ , 其中  $n$  为整数, 请在空格处填写合适的语句完成程序功能。

```
#include <iostream>
using namespace std;
① double power(T x, int n) {
 if(x==0) return 0;
 if(n==0) return 1;
 ②
 for(int i=0;i<abs(n);i++)
 ③
 if(n<0)
 return 1.0/powerx;
 else
 ④
}
```



3. 分析下面的程序，写出程序运行的结果

```
#include<iostream>
using namespace std;
class A {
public:
 void direct() { f1(); f2(); }
 virtual void f1(){ cout<<"A-F1"<<endl; }
 void f2(){ cout<<"A-F2"<<endl; }
};
class B : public A {
public:
 void f1(){ cout<<"B-F1"<<endl; }
 void f2(){ cout<<"B-F2"<<endl; }
};
void main() {
 A *p=new B;
 p->direct();
}
```

4. 分析下面的程序，写出程序运行的结果

```
#include <iostream>
using namespace std;
class S {
public:
 ~S(){ cout<<"S"<<"\t"; }
};
char fun0() {
 S s1;
 throw('T');
 return '0';
}
void main() {
 try {
 cout << fun0() << "\t";
 } catch(char c) {
 cout << c << "\t";
 }
}
```

|    |     |
|----|-----|
| 得分 | 评阅人 |
|    |     |

四、编程题。（共 3 题，每小题 10 分，共 30 分）

1. 定义一个 shape 抽象类，派生出 Rectangle 类和 Circle 类，计算各派生类对象的面积 Area()，并在 Rectangle 类中重载关系运算符 == 来比较两个 Rectangle 是否相等。

2. 设计一个类模板，其中包括数据成员  $T\ a[n]$  以及对其进行排序的成员函数 `sort()`，模板参数  $T$  可实例化成字符串。

线

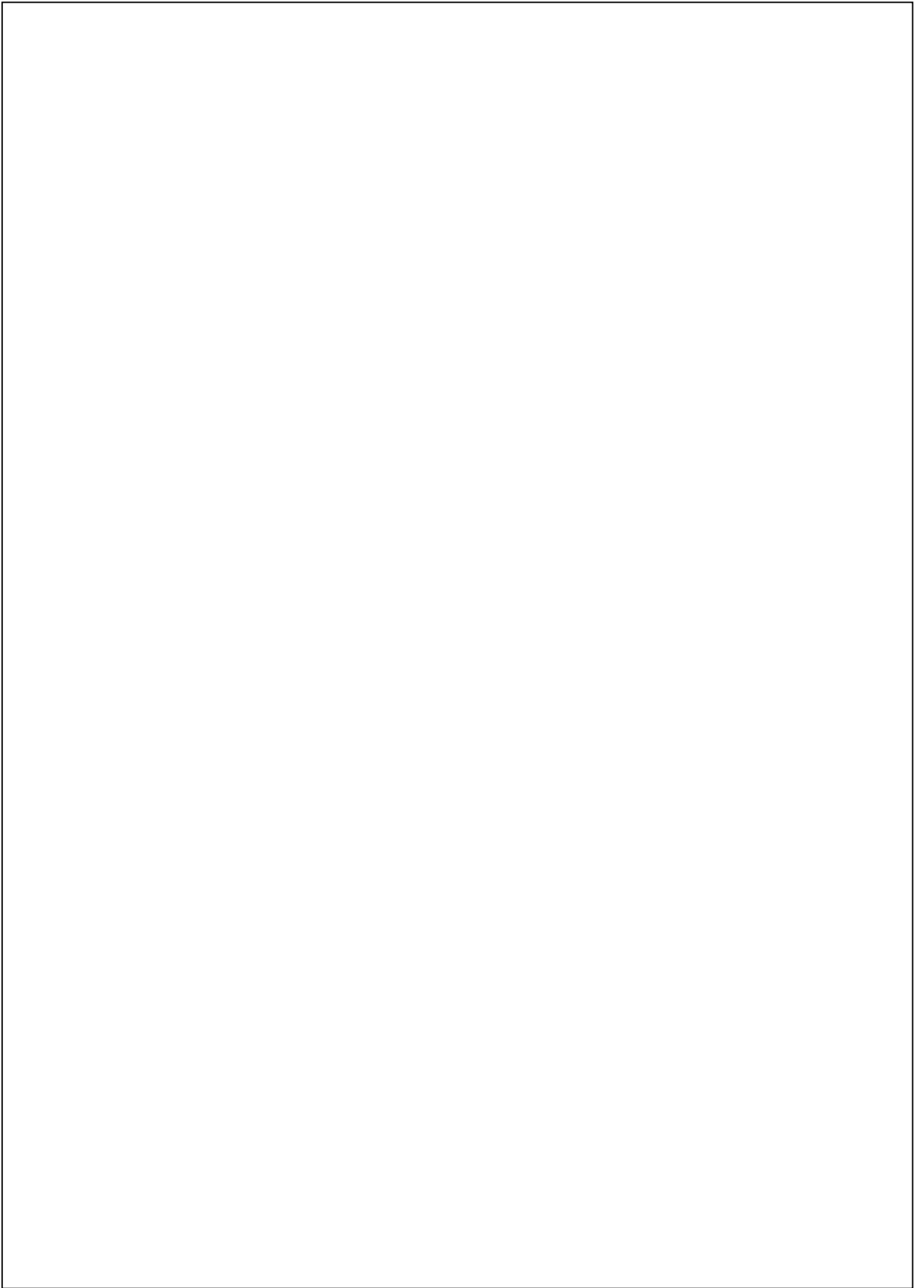
封

密

3. 定义一个 **Student** 类,其中含学号、姓名、成绩数据成员。建立若干个 **Student** 类对象,将它们保存到文件 **Record.dat** 中 (写文件), 然后显示文件中的内容 (读文件)。(部分主函数如下)

```
void main() {
 Student S1[3]={ Student(200907001,"Tom",76),
 Student(200907002,"John",82),
 Student(200907003,"Wilson",95) };

}
```



( ) 6. 以下说法正确的是:

- A. 成员对象都是用无参构造函数初始化的    B. 封闭类中成员对象的构造函数先于封闭类的构造函数被调用    C. 封闭类中成员对象的析构函数先于封闭类的析构函数被调用    D. 若封闭类有多个成员对象, 则它们的初始化顺序取决于封闭类构造函数中的成员初始化列表

( ) 7. 下面说法哪个不正确?

- A. 静态成员函数内部不能访问同类的非静态成员变量, 也不能调用同类的非静态成员函数    B. 非静态成员函数不能访问静态成员变量    C. 静态成员变量被所有对象所共享    D. 在没有任何对象存在的情况下, 也可以访问类的静态成员

( ) 8. 如果将[]运算符重载成一个类的成员函数, 则该重载函数有几个参数?

- A. 0    B. 1    C. 2    D. 3

( ) 9. 如何区分自增运算符重载的前置形式和后置形式?

- A. 重载时, 前置形式的函数名是 ++ operator, 后置形式的函数名是 operator ++    B. 后置形式比前置形式多一个 int 类型的参数    C. 无法区分, 使用时不管前置形式还是后置形式, 都调用相同的重载函数    D. 前置形式比后置形式多了一个 int 类型的参数

( ) 10. 派生类和基类有同名同参数表的成员函数, 这种现象:

- A. 叫重复定义, 是不允许的    B. 叫函数的重载    C. 叫覆盖。在派生类中基类的同名函数就没用了    D. 叫覆盖。体现了派生类对从基类继承得到的特点的修改

( ) 11. 以下关于友元的说法哪个是不正确的?

- A. 一个类的友元函数中可以访问该类对象的私有成员    B. 友元类关系是相互的, 即若类 A 是类 B 的友元, 则类 B 也是类 A 的友元    C. 在一个类中可以将另一个类的成员函数声明为友元    D. 类之间的友元关系不能传递

( ) 12. 以下基类中的成员函数, 表示纯虚函数的是

- A. virtual void vf(int);    B. void vf(int)=0;    C. virtual void vf()=0;    D. virtual void yf(int){}

( ) 13. 类模板的使用实际上是将类模板实例化成一个

- A. 函数    B. 对象    C. 类    D. 抽象类

- ( ) 14. 当一个类的某个函数被说明为 `virtual` 时, 该函数在该类的所有派生类中  
A. 都是虚函数      B. 只有被重新说明      C. 只有被重新说明      D. 都不是虚函数  
时才是虚函数      为 `virtual` 时才是  
虚函数

- ( ) 15. 如下程序的运行结果是哪一个?

```
#include <iostream>
using namespace std;
class A {
public:
 A() { cout << "A"; }
};
class B { public: B() { cout << "B"; } };
class C : public A {
 B b;
public:
 C() { cout << "C"; }
};
int main() { C obj; return 0; }
```

- A. CBA      B. BAC      C. ACB      D. ABC

|    |     |
|----|-----|
| 得分 | 评阅人 |
|    |     |

## 二、简答题。(共 5 题, 每小题 5 分, 共 25 分)

1. 继承与组合的区别与联系分别是什么? 请举例说明。

2. 什么是浅拷贝？什么是深拷贝？二者有何异同？请画图举例说明。

3. 什么是抽象类？抽象类有何作用？可以声明抽象类的对象吗？为什么？



线

封

密

4. 什么是多态？请举例说明在面向对象的程序设计中使用多态的好处。

5. 请依此解释 C++ STL 中的容器、迭代器和算法的概念。

|    |     |
|----|-----|
| 得分 | 评阅人 |
|    |     |

### 三、程序分析题。(共 2 题，第一小题 8 分、第二小题 7 分，共 15 分)

1. 分析下面的程序，写出程序运行的结果。

```
#include<iostream>
using namespace std;
class Demo {
 int id;
public:
 Demo(int i) {
 id = i;
 cout << "id=" << id << " constructed" << endl;
 }
 ~Demo() {
 cout << "id=" << id << " destructed" << endl;
 }
};
Demo d1(1);
void Func()
{
 static Demo d2(2);
 Demo d3(3);
 cout << "func" << endl;
}
int main () {
 Demo d4(4);
 d4 = 6;
 cout << "main" << endl;
 { Demo d5(5);
 }
 Func();
 cout << "main ends" << endl;
 return 0;
}
```

2. 分析下面的程序，写出程序运行的结果

```
#include <iostream>
using namespace std;
class S {
public:
 ~S(){ cout<<"S"<<"\t"; }
};
char fun0() {
 S s1;
 throw('T');
 return '0';
}
void main() {
 try {
 cout << fun0() << "\t";
 } catch(char c) {
 cout << c << "\t";
 }
}
```

|    |     |
|----|-----|
| 得分 | 评阅人 |
|    |     |

#### 四、编程题。（共 2 题，每小题 15 分，共 30 分）

1. 定义一个 Shape 类，派生出 Rectangle 类和 Circle 类，计算各派生类对象的面积 Area()，并在 Rectangle 类中重载关系运算符 == 来比较两个 Rectangle 是否相等。

线

封

密

2. 定义一个 **Student** 类, 其中含学号、姓名、成绩数据成员。建立若干个 **Student** 类对象, 将它们保存到文件 **Record.dat** 中 (写文件), 然后显示文件中的内容 (读文件)。(部分主函数如下)

```
void main() {
 Student S1[3]={Student(200907001,"Tom",76),Student(200907002,"John",82),
 Student(200907003,"Wilson",95) };

}
```

