

# 上海海事大学试卷

2021—2022 学年第 1 学期期末考试

《 面向对象程序设计 》(B 卷)

班级                      学号                      姓名                      总分

题 目	一	二	三	四				
得 分								
阅卷人								

## 考生诚信考试承诺书

我承诺：自觉遵守上海海事大学考场规则，服从监考人员的监督管理，不做违背考试纪律的任何事情，做到诚信考试。如有违反任何考试规定，自愿接受上海海事大学有关条款处理（直至开除）。

考生签名：\_\_\_\_\_

日期：\_\_\_\_\_年\_\_\_\_月\_\_\_\_日

注意：请将所有试题的答案写到答题纸上。

一、选择题（共 10 题，每题 2 分，共 20 分）。

1. 下列运算符中，( )是在 C++中可以重载的运算符。

- A. ?:
- B. <<
- C. ::
- D. .\*

2. 下列说法不正确的是 ( )

- A. 一个类可以有多个构造函数
- B. 函数形参为类类型，则实参向行参传值时调用拷贝构造函数
- C. 析构造函数不可以重载
- D. 带参数的构造函数不可能是默认构造函数

3. class Time 中，要进行后缀自增运算符重载，则成员函数声明应写成( )。

- A. Time operator++();
- B. Time operator++( ) int;

- C. Time operator++( int);
- D. Time operator++( ) const;

4. 关于静态成员的描述，( ) 是正确的。

- A. 静态成员函数可以使用 this 指针
- B. 静态数据成员可以类体内初始化
- C. 静态数据成员不可以被类的对象来访问
- D. 静态数据成员可以直接用类名来访问

5. class Time{

```
public:  Time(int y,int m,int d):year(y),month(m),day(d){}
```

```
    int year,month,day;
```

```
};
```

```
Time t1(2021, 12 25), t2;
```

```
Time *const ptr=&t1;
```

则下列语句正确的是( )。

- A. ptr1=&t2;
- B. ptr1.year=2022;
- C. t1.year=2022;
- D. 以上都不对

6. 采用函数重载的目的是 ( )

- A. 节省空间
- B. 提高程序的运行速度
- C. 使用方便，提高可读性
- D. 实现代码共享

7. 下面类的定义，有( )处错误

```
class myclass{  
    int i;  
  
public:  
    void myclass(int v);  
    ~ myclass (int Value);  
}
```

- A. 1
- B. 2
- C. 3
- D. 4

8. 用 new 运算符创建一个含有 10 个元素的一维 Time 类对象数组，下列语句中正确的

是( )。

- A. Time \*p=new Time(10);
- B. Time \*p=new Time;
- C. Time \*p=new t[10];
- D. Time \*p=new Time[10];

9. 下面对类是私有成员的访问, ( )是正确的。

- A. 在子类的成员函数中
- B. 在本类的成员函数中
- C. 通过该类的对象访问
- D. 通过子类的对象访问

10. 下列说法正确的是 ( )

- A. 常对象可以调用该对象的非常成员函数
- B. 常数据成员的值必须要在类体外初始化
- C. 常对象中的成员函数会自动成为常成员函数
- D. 常成员函数可以引用非 const 数据成员

## 二、判断题 (共 5 题, 每题 2 分, 共 10 分)。

- 1. 子类对象不能使用父类的私有成员, 是因为子类对象没有继承到父类的私有成员。
- 2. 在 void fun(const Student &s)函数中, 将形参定义为 Student 类型常引用, 则不能通过引用变量 s 修改该对象的值。
- 3. 类中的析构函数是用来释放该类对象的。
- 4. 派生类的对象可以赋值给基类形参变量。
- 5. 在定义派生类的构造函数时, 必须要调用基类的构造函数。

## 三、看程序, 写结果 (共 5 题, 每题 6 分, 共 30 分)。

1. 程序如下:

```
class Myclass{
public:
    Myclass(){cout<<"Constructor Called!"<<endl; j+=10;}
    void print(){cout<<"The value of j is:"<<j<<endl;}
    ~Myclass(){cout<<"Destructor Called!"<<endl;}
private:
    static int j;
};
```

```

int Myclass::j=0;
int main(){
    Myclass *p=new Myclass[3];
    for(int i=0;i<3;i++)
        p->print();
    delete[] p;
    return 0;
}

```

运行结果：

2. 程序如下：

```

class Base{
public:
    void who(){cout<<"class Base"<<endl;}
};
class A0:public Base{
protected:
    void who(){cout<<"class A0"<<endl;}
};
class A1:private A0{
public:
    void who(){cout<<"class A1"<<endl;}
};
int main(){
    Base obj1,*p;
    A0 obj2;
    A1 obj3;
    p=&obj1;
    p->who();
    p=&obj2;
    p->who();
    obj1.who();
    obj3.who();
    return 0;
}

```

运行结果：

3. 程序如下：

```

class A{
public:
    A(char i){cout<<i<<endl;}
}

```

```

    ~A() { cout<<"A"<<endl; }
};
class B:virtual public A{
public:
    B(char i,char j):A(i),b(j){ cout<<j<<endl; }
    ~B(){ cout<<"B"<<endl; }
private:
    char b;
};
class C:virtual public A{
public:
    C(char i,char k):A(i),c(k){ cout<<k<<endl; }
    ~C(){ cout<<"C"<<endl; }
private:
    char c;
};
class D:public B,public C{
public:
    D(char i,char j,char k,char t):B(i,j),C(i,k),A(i),d(t){ cout<<d<<endl; }
    ~D(){ cout<<"D"<<endl; }
private:
    char d;
};
int main(){
    D obj('a','b','c','d');
    return 0;
}
运行结果为:

```

4. 程序如下:

```

class Rect{
public:
    Rect(){ length=0;width=0; }
    Rect(int l,int w):length(l),width(w){ }
    ~Rect(){ }
    void display(){ cout<<"Area:"<<length*width<<endl; }
    Rect operator*(int scale){
        Rect t;
        t.length=length*scale;
        t.width=width*scale;
        return t;
    }
private:

```

```

    int length;
    int width;
};
int main() {
    Rect r1,r2(2,3);
    r1=r2*3;
    r1.display();
    r2.display();
    return 0;
}

```

运行结果:

4. 程序如下:

```

class Coordinate{
public:
    Coordinate(int a=0,int b=0):x(a),y(b){ }
    void display(){cout<<" "<<x<<" "<<y<<" "<<endl;}
    friend Coordinate operator++(Coordinate &p);
    friend Coordinate operator++(Coordinate &p,int);
private:
    int x,y;
};
Coordinate operator++(Coordinate &p){
    ++p.x;
    ++p.y;
    return p;
}
Coordinate operator++(Coordinate &p,int){
    ++p.x;
    ++p.y;
    return p;
}
int main(int argc, char** argv) {
    Coordinate point(1,2);
    point.display();
    ++point;
    point.display();
    point++;
    point.display();
    return 0;
}

```

运行结果:

#### 四、编程题（共 4 题，每题 10 分，共 40 分）。

1. 定义一个商品类（Product），包含私有数据成员商品名称（name）、价格（price）和数量（num），包含对外接口成员函数 setdata（）和 output（）。在主函数中定义一个商品类对象数组（Product pd[10]），完成该对象数组的赋值和信息输出。

2. 定义一个课程类（Course），包含数据成员课程号（ID）和分数（score），完成对应的构造函数定义。再定义一个学生类（Student），包含数据成员学号（num）、姓名（name）和成绩（grade），其中成绩(grade)是课程类类型，定义学生类相应的构造函数和输出函数。主函数中自拟数据进行测试。

3. 一个 3\*3 的矩阵类（Matrix）定义如下，要求完成相应的构造函数定义，重载“+”运算符，完成两个矩阵的加法，重载输出运算符能够按矩阵形式输出一个矩阵。主函数中自拟数据进行测试。

```
class Matrix{
private:
    int element[3][3];
public:
    Matrix(int a[3][3]);
    friend Matrix operator+(Matrix &,Matrix &);
    friend ostream& operator<<(ostream &,const Matrix &);
};
```

4. 定义一个抽象动物类（Animal），包含纯虚函数 sound（）和虚函数 display（）。由其派生出一个小鸡类（Chick）和一个小鸭类（Duck），要求在小鸡类中和小鸭类中完成两个基类成员函数的重定义，小鸡的 sound（）响应为“Peep”，小鸭的 sound（）响应为“Quack”，display（）函数输出各自的类型名称。在主函数中对 sound（）函数实现动态关联测试，对 display（）实现静态关联测试。

答题纸

班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_ 总分 \_\_\_\_\_

一、选择题（共 10 题，每题 2 分，共 20 分）。

题号	1	2	3	4	5	6	7	8	9	10
答案										

二、判断题（共 5 题，每题 2 分，共 10 分）。

题号	答案（对或错）
1	
2	
3	
4	
5	

三、看程序，写结果（共 5 题，每题 6 分，共 30 分）。

1.

2.

3.



4.

5.

四、编程题（共 4 题，每题 10 分，共 40 分）。

