

座位号：

杭州电子科技大学学生考试卷（ A ） 卷

考试课程	C++面向对象程序设计(甲)	考试日期	2015 年 6 月 16 日		成 绩		
课程号	A0507070	教师号		任课教师姓名			
考生姓名		学号 (8 位)		年 级		专 业	

注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交

一、判断题（对的打“√”，错的打“×”）（每题 1 分，总计 10 分）

题 号	1	2	3	4	5
答 案					
题 号	6	7	8	9	10
答 案					

二、单项选择题（每题 2 分，总计 20 分）

题 号	1	2	3	4	5
答 案					
题 号	6	7	8	9	10
答 案					

三、程序填空题（每空 2 分，总计 10 分）

- ①_____
- ②_____
- ③_____
- ④_____
- ⑤_____

四、阅读程序题（总计 30 分）

1. 阅读该程序，给出程序的输出结果。（6 分）

2. 阅读该程序，给出程序的输出结果。（8 分）

3. 阅读该程序，给出程序的输出结果。（8 分）

4. 阅读该程序，给出程序的输出结果。（8 分）

座位号：

五、编程题（总计 30 分）

- 1. 按照要求，编写程序（10 分）

座位号：

2. 按照要求，编写程序（20 分）

杭州电子科技大学学生考试卷（ A ）卷

考试课程	C++面向对象程序设计(甲)		考试日期	2015 年 6 月 16 日		成 绩	
课程号	A0507070	教师号		任课教师姓名			
考生姓名		学号 (8 位)		年 级		专 业	

注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交

一、判断题（对的打“√”，错的打“×”）（每题 1 分，总计 10 分）

- 编译 C++源程序时，出现了警告错（Warning）也可以生成可执行文件。
- 面向对象方法具有封装性、继承性和多态性。
- 用来给数组进行初始化的初始值表内的数据项的个数必须小于等于数组元素个数。
- 指针是变量，它具有的值是某个变量或对象的地址值，它还具有一个地址值，这两个地址值是相等的。
- 在 C++中，可以通过 ifstream 类对象来建立一个输出文件流。
- 创建对象时系统自动调用相应的构造函数为对象初始化，没有相应的构造函数时，系统会自动生成。
- 常成员指的是类体内使用 const 关键字说明的常数据成员和常成员函数。
- 任何对象都可以使用 delete 运算符来释放。
- 基类中成员在派生类中都是可以访问的。
- 使用成员函数方法和使用友元函数的方法所定义的重载运算符函数的参数个数是不相同的。

二、单项选择题（每题 2 分，总计 20 分）

- 下列关于创建一个 int 型变量的引用，正确的是（ ）。
A. int a(3), &ra=a; B. int a(3), &ra=&a;
C. double d(3.1); int &rd=d; D. int a(3), ra=a;
- 当一个函数没有返回值时，该函数类型应说明为（ ）。
A. void B. int
C. 无 D. 任意
- 下列有关类的静态成员描述正确的是（ ）。
A. 可以在类内初始化
B. 不能被类的对象调用
C. 不能受 private 修饰符的作用
D. 可以直接用类名调用
- 执行下列说明语句，其调用构造函数的次数为（ ）。

A a[5], *p[2]; 其中，A 是一个类名

- A. 5 B. 6
C. 7 D. 10

- 下列关于常成员的描述中，错误的是（ ）。
A. 常成员是用关键字 const 说明的
B. 常成员有常数据成员和常成员函数两种
C. 常数据成员的初始化是在类体内定义它时进行的
D. 常数据成员的值是不可以改变的
- 已知：const A a; 其中 A 是一个类名，指向常对象指针的表示为（ ）。
A. const *A pa; B. const A *pa;
C. A *const pa; D. const *pa A;
- 下列关于 this 的描述中，错误的是（ ）。
A. this 是一个由系统自动生成的指针
B. this 指针是指向对象的
C. this 指针在用对象引用成员函数时系统创建的
D. this 指针只能隐含使用，不能显式使用
- 在重载一个运算符时，如果其参数表中有一个参数，则说明该运算符是（ ）。
A. 一元成员运算符 B. 二元成员运算符
C. 一元友元运算符 D. B 和 C 都可能
- 下列运算符中，不可以重载的是（ ）。
A. && B. &
C. [] D. ::
- 下列关于抽象类的描述中，错误的是（ ）。
A. 抽象类中至少应该有一个纯虚函数 B. 抽象类可以定义对象指针和对象引用
C. 抽象类通常用作类族中最顶层的类 D. 抽象类的派生类必定是具体类

三、程序填空题（每空 2 分，总计 10 分）

- 完成下列类的定义。
class A
{
 int a;
public:
 A() { a=0; }
 ① int Geta(② &m);
};
int Geta(③ &m)
{
 return m.a;
}

2. 请在下面程序的横线处填上适当内容，以使程序完整，并使程序的输出为：

```

    B1 called
    B2 called
#include<iostream>
using namespace std;
class A
{
public:
    _____④;
};

class B1:public A
{
public:
    void display() { cout<<"B1 called"<<endl; }
};

class B2:public A
{
public:
    void display() { cout<<"B2 called"<<endl; }
};

void show(_____⑤)
{
    p->display();
}

void main()
{
    B1 b1;
    B2 b2;
    A* p[2]={&b1,&b2};
    for(int i=0;i<2;i++)
        show(p[i]);
}
```

四、阅读程序题（总计 30 分）

1. 阅读该程序，给出程序的输出结果。（6 分）

```

#include <iostream>
using namespace std;
class A
{
public:
    virtual void fun()
    {
        cout<<"A::fun() called.\n";
    }
};
class B:public A
{
    void fun()
    {
        cout<<"B::fun() called.\n";
    }
};
void ffun(A *pa)
{
    pa->fun();
}
void main()
{
    A *pa=new A;
    ffun(pa);
    B *pb=new B;
    ffun(pb);
}
```

2. 阅读该程序，给出程序的输出结果。（8 分）

```

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

class MyException {
public :
    MyException(string msg) { err_msg = msg; }
    void ShowErrorMsg() { cerr<<err_msg<<endl; }
```

```

        ~MyException() { }
    private:
        string err_msg;
};
void KindsOfException(int type) throw (MyException,int,double) {
    switch(type) {
        case 0:
            throw MyException("Exception! Type of Class");
            break;
        case 1:
            throw 2015;
            break;
        case 2:
            throw 1.23;
            break;
        default:
            break;
    }
}
int main()
{
    int type[3] = {1, 2, 0};
    int e;
    for ( int i = 0; i < 3; i++ )
    {
        e = type[i];
        try {
            KindsOfException(e);
        }
        catch(MyException myE) {
            myE.ShowErrorMsg();
        }
        catch (double d) {
            cerr<<"double"<<d<<endl;
        }
        catch(int i) {
            cerr<<"Exception! Type of Int -->"<<i<<endl;
        }
    }
}

```

```

        return 0;
    }

3. 阅读该程序，给出程序的输出结果。（8分）
#include <iostream>
using namespace std;
class A
{
public:
    A(int i)
    {
        a=i;
    }
    A()
    {
        a=0;
        cout<<"Default constructor called."<<a<<endl;
    }
    ~A()
    {
        cout<<"Destructor called."<<a<<endl;
    }
    void Print()
    {
        cout<<a<<endl;
    }
private:
    int a;
};
void main()
{
    A a[4],*p;
    int n=1;
    p=a;
    for(int i=0;i<4;i++)
        a[i]=A(++n);
    for(i=0;i<4;i++)
        (p+i)->Print();
}

```

4. 阅读该程序，给出程序的输出结果。（8分）

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

int main()
{
    int i = 0, j = 0;
    int count = 0;
    vector< vector<int> > Array;
    vector<int> line;
    for( j = 0; j < 3; j++ )
    {
        Array.push_back(line);
        for ( i = j + 3; i < j + 7; i++ )
        {
            Array[j].push_back(i);
        }
    }
    for( j = 0; j < 3; j++ )
    {
        for( i = 0; i < Array[j].size(); i++ )
        {
            cout << Array[j][i] << " ";
        }
        cout << endl;
    }
    for( j = 0; j < 3; j++ )
    {
        if ( find(Array[j].begin(), Array[j].end(), 6) != Array[j].end() )
            count++;
    }
    cout << count << endl;

    return 0;
}
```

五、编程题（总计 30 分）

- 1. （10 分）按下列要求编程，即实现相关类，并在主函数中使用这些类。
自行车（Bicycle）和汽车（Motorcar）都是车辆（Vehicle），它们有共同的属性最大速度（maxSpeed）和重量（weight），也有各自不同的特性，比如自行车的高度（height）和汽车的座位数（seatNum）。现有不同类型车辆若干（假定为 3），将其输入并放入一个指针数组，每个车辆需要设置其属性。输入后分别显示各自属性（即自行车和汽车分别显示各自属性）。
- 2、（20 分）编程实现一个存储任意类型的集合类（即模板类），要求如下：
 - (1) 集合中的元素用动态链表存放
 - (2) 提供必要的构造函数、移动构造函数、赋值运算符重载函数等
 - (3) 能显示集合中元素个数
 - (4) 能显示集合中的元素
 - (5) 能向集合中增加元素（不能出现重复）
 - (6) 能从集合中删除元素
 - (7) 能实现两个集合的并运算（+）
 - (8) 能实现两个集合的交运算（*）
 - (9) 能实现两个集合的差集(-)
 - (10) 能使用<<运算符对集合类对象进行输出