

# 浙江大学 2008 - 2009 学年秋冬季学期

## 《面向对象程序设计》课程期末考试试卷

开课学院：计算机，考试形式：闭卷，允许带 NULL 入场

考试时间：2009 年 1 月 7 日, 所需时间：120 分钟，任课教师 \_\_\_\_\_

考生姓名：\_\_\_\_\_ 学号：\_\_\_\_\_ 专业：\_\_\_\_\_

题序	一	二	三	四	五	六	七	八	总分
得分									
评卷人									

### 1、Write the output of the code below (32%)

1)

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

```
template <typename T>
```

```
void fun(T &x, T &y)
```

```
{
```

```
    T temp;
```

```
    temp = x; x = y; y = temp;
```

```
}
```

```
void main()
```

```
{
```

```
    int i,j;
```

```
    int *pi = &i, *pj = &j;
```

```
    i = 10;
```

```
    j = 20;
```

```
    fun(i,j);
```

```
    cout << "i = " << i << "\t" << "j = " << j << endl;
```

```
    fun(pi, pj);
```

```
    cout << "i = " << i << "\t" << "j = " << j << endl;
```

```
}
```

2)

```
#include <iostream.h>
```

```
class complex{
```

各科复习资料，蓝田益汇图文31

**浙江大学 2008 - 2009 学年秋冬季学期**  
**《面向对象程序设计》课程期末考试试卷**

开课学院：计算机，考试形式：闭卷，允许带 NULL 入场

考试时间：2009 年 1 月 7 日，所需时间：120 分钟，任课教师                     

考生姓名：                      学号：                      专业：                     

题序	一	二	三	四	五	六	七	八	总分
得分									
评卷人									

**1、Write the output of the code below (32%)**

1)

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

```
template <typename T>
void fun(T &x, T &y)
```

```
{
    T temp;
    temp = x; x = y; y = temp;
}
```

```
void main()
```

```
{
    int i,j;
    int *pi = &i, *pj = &j;
    i = 10;
    j = 20;
    fun(i,j);
    cout << "i = " << i << "\t" << "j = " << j << endl;

    fun(pi, pj);
    cout << "i = " << i << "\t" << "j = " << j << endl;
}
```

2)

```
#include <iostream.h>
```

```
class complex{
```

各科复习资料，蓝田益汇图文31

```

private:
    double real_part;
    double imaginary_part;
public:
    complex(double real = 0 ,double imag= 0 ):
    real_part(real),imaginary_part(imag)
    {
        cout << '(' << real_part;
        if (imaginary_part != 0.0)
        {
            cout << '+' << imaginary_part << 'i';
        }
        cout << ')' << endl;
    }
    const complex operator+(const complex &com) ;
    friend const complex operator+(const complex& left, const complex& right);
};

const complex complex::operator+(const complex& com)
{
    cout << "calling member function" << endl;
    return complex(real_part+com.real_part,
    imaginary_part+com.imaginary_part);
}

const complex operator+(const complex& left, const complex& right)
{
    cout << "calling friend global function" << endl;
    return complex(left.real_part+right.real_part,
    left.imaginary_part+right.imaginary_part);
}

void main()
{
    complex a(1,2), b(0.1,0.2), c;
    c = a + b;
}

```

3)

```

#include <iostream>
using namespace std;

```

```

void f(int i)
{

```



```

        cout << "f(int):" << i << endl;
    }
    void f(float i)
    {
        cout << "f(float):" << i << endl;
    }
    template <class T>
    void f(T i)
    {
        cout << "f(T)" << i << endl;
    }
    void main()
    {
        f(1);
        f(1.0);
    }
4)
#include <iostream>
using namespace std;
int count;

class myCla
{
private:
    char cc;
public:
    myCla(char ch)
    {
        ++count;
        cc = ch;
        cout << "constructor:count=" << count << ",ch=" << ch << endl;
    }
    ~myCla()
    {
        --count;
        cout << "destructor:count=" << count << ",cc=" << getcc() << endl;
    }
    char getcc()
    {
        return cc;
    }
};

myCla globalG('G');

```

```

int main()
{
    myCla autoA('A');
    for(int i=1;i<=2;i++)
    {
        cout<<"——beging block"<<endl;
        myCla autoB('B');
        static myCla staticS('S');
        cout<<"——end block"<<endl;
    }
    cout<<"——end main"<<endl;
    return 0;
}

```

5)

```

#include <iostream>
using namespace std;

```

```

class A{
    static int m;
    int n;
public:
    A(int m,int n)
    {
        this->m = m;
        this->n = n;
    }
    void print()
    {
        cout << m << "—" << n << endl;
    }
};

```

```

int A::m;

```

```

void main()

```

```

{
    A a1(3,4);
    A a2(5,6);
    a1.print();
    a2.print();
}

```

6)

```

int aa1=53,aa2=69;
void f(int a1,int &a2)
{

```

```

        a2 = a1;
        a1 += a2;
        cout << aa1 << aa2 << endl;
        aa2 -= 7;
        a2++;
    }
    void main()
    {
        f(aa1,aa2);
        cout << aa1 << aa2 << endl;
    }
7)
#include<iostream>
using namespace std;
class B{
public:
    void f()
    {
        cout << "bf ";
    }
    virtual void vf()
    {
        cout << "bv ";
    }
    void ff()
    {
        vf();
        f();
    }
    virtual void vff()
    {
        vf();
        f();
    }
};
class D: public B{
public:
    void f()
    {
        cout<<"df ";
    }
    void ff()
    {
        f();
    }
};

```

```

        vf();
    }
    void vf()
    {
        cout<<"dvf ";
    }
};

```

```

void main()
{
    D d;
    B *pB = &d;
    pB->f();
    pB->ff();
    pB->vf();
    pB->vff();
}

```

8)

```

#include <iostream>
using namespace std;

```

```

template <class T>
void print(const T &val)
{
    cout << val;
}

```

```

template <>
void print(const float &f_val)
{
    int i = f_val;
    cout << i;
}

```

```

int main()
{
    print("Today's temperature: ");
    print((float)33.4);

    return 0;
}

```

**2. Please correct the following programs (point out the errors and correct them) (12%)**

各科复习资料，蓝田益汇图文36

```

        vf();
    }
    void vf()
    {
        cout<<"dvf ";
    }
};

```

```

void main()
{

```

```

    D d;
    B *pB = &d;
    pB->f();
    pB->ff();
    pB->vf();
    pB->vff();
}

```

8)

```

#include <iostream>
using namespace std;

```

```

template <class T>
void print(const T &val)
{
    cout << val;
}

```

```

template <>
void print(const float &f_val)
{
    int i = f_val;
    cout << i;
}

```

```

int main()
{
    print("Today's temperature: ");
    print((float)33.4);

    return 0;
}

```

**2. Please correct the following programs (point out the errors and correct them) (12%)**

各科复习资料，蓝田益汇图文36



```

1)
class Foo {};
void fun(Foo f) {}
void fun(Foo& f) {}

```

```

main()
{
    Foo ff;
    func(ff);
    func(&ff);
}

```

```

2)
class A
{
    static int k;
public:
    static void SetK(int kk){
        k = kk;
    }
    static int GetK()const{
        return k;
    }
};

```

```

int k;
void main()
{
    A a;
    a.SetK(333);
    cout << a.GetK() << endl;
}

```

```

3)
class Str
{
    char *s;
public:
    Str(char *p){
        s = new [strlen(p)+1];
        strcpy(s,p);
    }
    ~Str(){
        delete s;
    }
    char operator[](int i){

```

```

        return s[i];
    }
};
Str First(const Str &s){
    cout << s[0] << endl;
    return s;
}
void main()
{
    Str s("Hi");
    Str s2 = First(s);
}

```

### 3、Fill in the blanks (26%)

1)The function template MaxMin() can find out the max and min of a two dimension array, row is first dimension of length and col is second dimension of length .

```

#include <iomanip.h>
_____ void MaxMin(T* array,int row,int col)
{
    T max = array[0],min = array[0];
    for(_____ ;i<row;i++)
        for( _____ ;j<col;j++)
        {
            if( _____ )
                max = array[i*row+j];
            if( _____ )
                min = array[i*row+j];
        }
    cout << "max=" << max << endl;
    cout << "min=" << min << endl;
}
void main()
{
    int ai[2][3]={{8,10,2},{14,4,6}};
    MaxMin( _____ );
}

```

2)The below function template can calculate the sum of two vectors.

```
#include <iostream.h>
```

```
template <class T>
```

```

    sum(_____, _____, _____)
{
    T* sum = _____;
    for(int i=0; i<size; i++)
        sum[i] = a[i]+b[i];
    return sum;
}

```

void main()

```

{
    int a[5]={1,2,3,4,5}, b[5]={10,20,30,40}, *p;
    double a2[5]={1.1,2.2,3.3,4.4,5.5}, b2[5]={10.6,20.7,30.8,40.9}, *p2;
    p = sum(a,b,5);
    p2 = sum(a2,b2,5);

    for(int i=0;i<5;i++)
        cout<<p[i]<<" "<<p2[i]<<endl;
        _____;
        _____;
}

```

#### 4、Program Design (30%)

Create a class named vehicle, which including data members: number of wheels and vehicle weight. Class car is derived from vehicle, having a private data member passenger\_load. Class truck is also derived from vehicle, and have two private data members named passenger\_load and payload. Each class contains its own function member to display its data. client can using these classes as follows:

```

void main ()
{
    vehicle *p;
    p = new car(4,2000,5);
    p->show ();
    delete p;
    p = new truck(10,8000,3,340000);
    p->show ();
    delete p;
}

```

output of above code:

```

Model:car
wheels: 4
weight: 2000 kg
passenger_load: 5

```

**Model:**truck  
**wheels:** 10  
**weight:**8000 kg  
**passenger\_load:** 3  
**payload:** 34000 kg

蓝田益汇图文