

# 浙江大学 2007 - 2008 学年春夏季学期

## 《面向对象程序设计》课程期中考试试卷

开课学院：计算机，考试形式：闭卷，允许带 NULL 入场

考试时间：2008 年 4 月 21 日，所需时间：45 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 (50%)

1)

```
#include<string>
#include<iostream>
using namespace std;
class Student
{
public:
    student(int n,string nam,char s)
    {
        num=n;
        name=nam;
        sex=s;
        cout<<"Constructor called."<<endl;
    }
    ~Student()
    {
        cout<<"Destructor called."<<endl;
    }
    void display()
    {
        cout<<"num: "<<num<<endl;
        cout<<"name: "<<name<<endl;
        cout<<"sex: "<<sex<<endl<<endl;
    }
private:
    int num;
    char name[10];
```

各科复习资料，蓝田益汇图文41

```

char sex;
};
int main( )
{
    Student stud1(10010,"Wang_li",'f');
    stud1.display( );
    Student stud2(10011,"Zhang_fun",'m');
    stud2.display( );
    return 0;
}

```

答案:

Constructor called.

num: 10010

name: Wang\_li

sex: f

Constructor called.

num: 10011

name: Zhang\_fun

sex: m

Destructor called.

Destructor called.

2)

```

#include <iostream>
using namespace std;

```

```

class A
{
public:
    A(int i = 0):m_i(i){};
    void display()
    {
        cout << m_i << endl;
    }
private:
    int m_i;
    friend void main();
};

```

```

void createA(A* pA)
{
    pA = new A(1);
}

```

```

A* createA()

```

```

{
    A a(2);
    return &a;
}

void createAOnStack()
{
    A a(3);
}

void main()
{
    A a;
    A* pA = &a;
    pA->display();
    createA(pA);
    pA->display();

    A* a2 = createA();
    cout << a2->m_i << endl;
    createAOnStack();
    cout << a2->m_i << endl;
}

```

答案:

0

0

2

3

3)

```

#include <iostream>
using namespace std;

```

```

class Name

```

```

{

```

```

public:

```

```

    Name(const char* first,const char* last)

```

```

    {

```

```

        strcpy(m_first,first);

```

```

        strcpy(m_last,last);

```

```

    }

```

```

    void display()const

```

```

    {

```

```

        cout << m_first << " " << m_last << endl;

```

各科复习资料, 蓝田益汇图文43

```

    }
    char* getFirstName()
    {
        return m_first;
    }
private:
    char m_first[20];
    char m_last[20];
};

```

```

class Person
{
public:
    Person(Name name):m_name(name){}
    Name getName()
    {
        return m_name;
    }
private:
    Name m_name;
};

```

```

void main()
{
    Person person(Name("Robin","Williams"));
    Name name = person.getName();
    char* firstName = name.getFirstName();
    firstName = "Tom";
    person.getName().display();
}

```

答案:

Robin Williams

4)

```
#include<iostream>
```

```
using namespace std;
```

```
class SuperMario
```

```
{
```

```
private:
```

```
    char* info;
```

```
public:
```

```
    SuperMario(char* name)
```

各科复习资料，蓝田益汇图文44



```

    }
    char* getFirstName()
    {
        return m_first;
    }
private:
    char m_first[20];
    char m_last[20];
};

```

```

class Person
{
public:
    Person(Name name):m_name(name){}
    Name getName()
    {
        return m_name;
    }
private:
    Name m_name;
};

```

```

void main()
{
    Person person(Name("Robin","Williams"));
    Name name = person.getName();
    char* firstName = name.getFirstName();
    firstName = "Tom";
    person.getName().display();
}

```

答案:

Robin Williams

4)

```
#include<iostream>
```

```
using namespace std;
```

```
class SuperMario
```

```
{
```

```
private:
```

```
    char* info;
```

```
public:
```

```
    SuperMario(char* name)
```

各科复习资料，蓝田益汇图文44

```

    {
        info=name;
        cout<<info<<" starts game!"<<endl;
    }
    ~SuperMario()
    {
        cout<<info<<" is Game Over!"<<endl;
    }
    void ShowInfo()
    {
        cout<<info<<" is trying to save Princess Peach."<<endl;
    }
};

class PaperMario : SuperMario
{
public:
    PaperMario(char* a):SuperMario(a){}

    void ShowInfo()
    {
        cout<<"Koopa is trying to save Princess Peach."<<endl;
    }
};

void main()
{
    PaperMario mario("Mario");
    SuperMario lugi("Lugi");
    mario.ShowInfo();
    lugi.ShowInfo();
}

```

答案:

**Mario starts game!**

**Lugi starts game!**

**Koopa is trying to save Princess Peach.**

**Lugi is trying to save Princess Peach.**

**Lugi is Game Over!**

**Mario is Game Over!**

**5)**

```
#include <iostream>
```

```
using namespace std;
```

```
class BaseClass
```

```

{
public:
    BaseClass(int i):m_i(i){}
    void f()
    {
        m_i += 8;
    }
    [virtual] void display()
    {
        cout << m_i << endl;
    }
private:
    int m_i;
};

class DerivedClass:public BaseClass
{
public:
    DerivedClass(int i,int j = 0):m_i(i),BaseClass(j){}
    void f()
    {
        m_i += 6;
    }
    void display()
    {
        cout << m_i << endl;
    }
private:
    int m_i;
};

void main()
{
    BaseClass* pB = new DerivedClass(5);
    pB->display();
    pB->f();
    pB->display();

    DerivedClass* pD = dynamic_cast<DerivedClass*>(pB);
    pD->display();
    pD->f();
    pD->display();
}

```

答案:

5  
5  
5  
11

## 2.Fill in the blanks (20%)

```
class String
{
    char *m_pStr;
    int m_len;
public:
    String(char *str){
        m_len = ::strlen(str);
        m_pStr = new char[_____];
        _____;
    }
    _____operator=(String str) const{
        return (_____==0);
    }
    char operaotr[] (int i) {
        return m_pStr[i];
    }
    _____operator<<(stream &out,String s);
};
_____operator<<(stream &out,String s)
{
    out<<_____;
    return out;
}
void main()
{
    String s1,s2("HELLO");
    s1="HELLO";
    if(s1==s2)
        cout<<"S1==S2\n";
    cout<<s1<<endl;
    cout<<s1[1];
}
```

## 3.Complete the following program (30%)

Stack is a kind of container which holds the property of "first in last out". Implement a push-down stack using linked list that satisfies the following declaration and usage.



```

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

class Stack
{
public:
    class Node //Linked List's Node
    {
    public:
        void* data;
        Node* next;
        Node(void* data, Node* next); //construction of Node
    }* head;
    Stack(); //construction of stack
    void push(void* data); //push a data into the stack
    void* top(); //get the data at stack top
    void* pop(); //get the data at stack top and remove it from the stack
    void cleanup(); //clean up the stack to size 0
};

void main()
{
    string strs[3] = {string("stack"),string("your"),string("test")};
    Stack s;
    for(int i = 0; i < 3; ++i)
        s.push(&strs[i]);
    string* str;
    while((str = (string*)s.pop()) != NULL)
        cout << *str << " ";
    s.cleanup();
}

//Example for answers
void* Stack::top()
{
    return (NULL == head) ? NULL : head->data;
}

```