

# 2007 级期末考试试卷（A 卷） 2008 年 6 月

## 一、单项选择(每空1分，共20分)

- 1、下列关于构造函数说法不正确的是（ ）。  
A) 构造函数必须与类同名  
B) 构造函数可以省略不写  
C) 构造函数必须有返回值  
D) 在构造函数中可以对类中的成员进行初始化
- 2、任意一个类，析构函数的个数最多是（ ）。  
A) 不限个数 (B) 1 (C) 2 (D) 3
- 3、友元的作用是（ ）。  
A) 增减成员函数的种类 B) 加强类的封装性  
C) 实现数据的隐藏 D) 提高程序的运行效率
- 4、在公有派生情况下，有关派生类对象和基类对象的关系，下列叙述不正确的是（ ）。  
A) 派生类的对象可以直接访问基类的所有成员  
B) 派生类的对象可以初始化基类的引用  
C) 派生类的对象不可以直接访问基类中的私有成员  
D) 派生类的对象的地址可以赋给指向基类的指针
- 5、下面对静态数据成员的描述中，正确的是（ ）。  
A) 静态数据成员可以在类体内进行初始化  
B) 静态数据成员不可以被该类的对象调用  
C) 静态数据成员不能受 `private` 控制符的作用  
D) 静态数据成员可以直接用类名访问
- 6、在类的定义中，用于为对象分配内存空间，对类的数据成员进行初始化并执行其他内部管理操作的函数是（ ）。  
A) 友元函数      B) 虚函数      C) 构造函数      D) 析构函数
- 7、下列对派生类的描述中错误的是（ ）。  
A) 派生类至少有一个基类

- B) 派生类可作为另一个派生类的基类
- C) 派生类除了包含它直接定义的成员外, 还包含其基类的成员
- D) 派生类对其继承的基类的成员访问权限保持不变

8、下列关于 **this** 指针的说法, 正确的是 ( )。

- A) **this** 指针一定指向常量型数据
- B) **this** 指向的数据不可更改
- C) 静态成员函数中也可以访问 **this** 指针
- D) **this** 指针本身可直接作为成员函数的返回值

9、下面关于类的成员函数描述不正确的是 ( )。

- A) 静态成员函数内可以直接访问类的非静态成员数据
- B) 静态成员函数内可以直接访问类的静态成员数据
- C) 非静态成员函数可以直接访问类的非静态成员数据
- D) 非静态成员函数可以直接访问类的静态成员数据

10、下列关于类和对象的叙述中, 错误的是 ( )。

- A) 一个类只能有一个对象
- B) 对象是类的具体实例
- C) 类是某一类对象的抽象
- D) 类和对象的关系是一种数据类型与变量的关系

11、有如下类声明:

```
class Sample { int x; };
```

则 **Sample** 类成员 **x** 是 **Sample** 类的 ( )。

- A) 公有数据成员 B) 保护数据成员
- C) 私有数据成员 D) 以上选项都不正确

12、在表达式 **x+y\*z** 中, **+** 是作为成员函数重载的运算符, **\*** 是作为非成员函数重载的运算符。下列叙述中正确的是 ( )。

- A) **operator+** 有两个参数, **operator\*** 有两个参数
- B) **operator+** 有两个参数, **operator\*** 有一个参数
- C) **operator+** 有一个参数, **operator\*** 有两个参数
- D) **operator+** 有一个参数, **operator\*** 有一个参数

13、下列关于虚函数的说明中，正确的是（ ）。

- A) 从虚基类继承的函数都是虚函数 B) 虚函数不得是静态成员函数
- C) 只能通过指针或引用调用虚函数 D) 抽象类中的成员函数都是虚函数

14、类模板的模板参数（ ）。

- A) 只可作为数据成员的类型 B) 只可作为成员的返回类型
- C) 只可作为成员函数的参数类型 D) 以上3项均可以

15、下列关于运算符重载的描述中，正确的是（ ）。

- A) 运算符重载可以改变操作数的个数
- B) 运算符重载可以改变运算符的优先级
- C) 运算符重载可以改变运算符的结合性
- D) 运算符重载可以使运算符实现特殊功能

16、已知 `outfile` 是一个输出流对象，要想将 `outfile` 的文件指针定位到当前位置之前321个字节处，正确的函数调用语句是（ ）。

- A) `outfile.seekp(321, ios::cur);` B) `outfile.seekp(321, ios::beg);`
- C) `outfile.seekp(-321, ios::beg);` D) `outfile.seekp(-321, ios::cur);`

17、下面描述中，错误的是（ ）。

- A) 公有继承时，基类中的 `public` 成员在派生类中仍是 `public` 的
- B) 公有继承时，基类中的 `private` 成员在派生类中仍是 `private` 的
- C) 公有继承时，基类中的 `protected` 成员在派生类中仍是 `protected` 的
- D) 私有继承时，基类中的 `public` 成员在派生类中是 `private` 的

18、派生类构造函数的调用顺序是先调用（ ）。

- A) 派生类的构造函数 B) 基类的构造函数
- C) 派生类的析构函数 D) 基类的析构函数

19、对于拷贝构造函数，正确的描述是（ ）。

- A) 在 C++ 语言中，如果不自定义类的拷贝构造函数，则每个类都有默认的拷贝构造函数
- B) 必须为每个类定义拷贝构造函数
- C) 如果要使用拷贝构造函数，则必须在类中先定义
- D) 当定义了类的构造函数时，如果要使用拷贝构造函数，则必须定义拷贝构造函数

20、对于语句 `cout<<endl<<x;` 中的各个组成部分，下列叙述中错误的是（ ）。

- A) “cout”是一个输出流对象 B) “endl”的作用是输出回车换行  
C) “x”是一个变量 D) “<<”称作提取运算符

## 二、判断题(每空1分，共10分)(注意答案 T 表示正确，F 表示错误)

- 1、在 C++ 中，允许从一个类派生出任意多个类，但不允许某个类同时从多个类派生。( )
- 2、抽象类虽然也是一种数据类型，但它不能建立对象，只能作为基类被其他类继承。( )
- 3、一个类模板可以先建立对象，再实例化为相应的模板类，然后使用这个模板类。( )
- 4、一个类的静态数据成员被该类的所有对象共享。( )
- 5、基类类型的指针可以指向其派生类对象。( )
- 6、任何类都要有不必提供参数的构造函数（默认缺省构造函数）。( )
- 7、私有继承中，对于基类中的所有成员，派生类的成员函数都不可直接访问。( )
- 8、在 C++ 中，可以声明虚构造函数和虚析构函数。( )
- 9、类中所提供的成员函数可以直接访问该类的私有成员。( )
- 10、抽象类不能定义对象，但可以声明抽象类的指针或引用。( )

## 三、读程序写结果(共36分)

- 1、写出下面程序的运行结果。(本题12分，每空3分)

```
#include<iostream.h>

class Sample
{
int x,y;
public:
Sample() {x=y=0;}
Sample(int a,int b) {x=a;y=b;}
~Sample()
{
if(x==y) cout<<"x=y"<<endl;
else cout<<"x!=y"<<endl;
}
```

```

void disp() { cout<<"x="<<x<<" ,y="<<y<<endl; }

};

void main()

{

Sample s1;

Sample *ps2=new Sample(2,3);

s1.disp();

ps2->disp();

delete ps2;

}

```

(1) \_\_\_\_\_

(2) \_\_\_\_\_

(3) \_\_\_\_\_

(4) \_\_\_\_\_

2、写出下面程序的运行结果。（本题6分，每空3分）

```

#include<iostream.h>

class Sample

{

public:

Sample(){}

Sample(int i){n=i;}

friend Sample operator-(Sample &,Sample &);

friend Sample operator+(Sample &,Sample &);

void disp(){cout<<"n="<<n<<endl;}

private:

int n;

};

Sample operator-(Sample &s1,Sample &s2)

{

```

```

int m=s1.n-s2.n;
Sample tmp(m);
return tmp;
}

Sample operator+(Sample &s1,Sample &s2)
{
int m=s1.n+s2.n;
Sample tmp(m);
return tmp;
}

void main()
{ Sample s1(10),s2(20),s3;
s3=s2-s1;
s3.disp();
s3=s2+s1;
s3.disp();
}

```

(5)\_\_\_\_\_

(6)\_\_\_\_\_

3、写出下面程序的运行结果。（本题12分，每空3分）

```

#include <iostream.h>

template <class Type> Type f(Type x[], int n)
{
int i,s=0;
Type tmp=x[0];
for(i=1; i<n; i++){
if(x[i]<tmp) tmp=x[i];
if(x[i]>x[0])s++;
}
}

```

```

cout<<"s="<<s<<endl;

return tmp;

}

void main() {

int a[]={1, 3, -6, 9, 2, 25, 44, -1, 51, 19};

char c[]={'b', 'T', 'a', 'A', 'Y', 'k'};

cout<<"a="<<f(a,8)<<endl;

cout<<"c="<<f(c,3)<<endl;

}

```

(7) \_\_\_\_\_

(8) \_\_\_\_\_

(9) \_\_\_\_\_

(10) \_\_\_\_\_

4、写出下面程序的运行结果。（本题6分，每空2分）

```

#include<iostream.h>

const int N=100;

class CStack

{ public:

CStack()

{ top=0;

cout<<"Hello"<<endl;

}

~CStack()

{ cout<<"Bye"; }

void push(int i);

int pop();

private:

int stack[N];

int top;

```

```

};

void CStack::push(int i)
{ if(top==N)
cout<<"Overflow";
else
{ top++;
stack[top]=i; }
}

int CStack::pop( )
{ int temp;
if (top==0)
{ cout<<"Underflow";
return 0;
}
else
{ temp=stack[top];
top--;
return temp;
}
}

void main()
{ CStack *ptr=new CStack;
ptr->push (10);
ptr->push(50);
cout <<ptr->pop( )<<" "<<endl;;
cout << "OK!"<<endl;
}

```

运行结果为:

(1)\_\_\_\_\_



(12) \_\_\_\_\_

(13) \_\_\_\_\_

#### 四、程序填空（每空2至3分，共34分）

1、下面的程序定义了两个类，一个是 **TennisClub**，其成员数据有俱乐部和会员名，另一个是 **Person**，其成员数据有姓名和年龄，类 **TennisClub** 的成员函数 **visit** 可以显示姓名、年龄、俱乐部名、会员名。其中 **TennisClub** 类是 **Person** 类的友类。请将程序补充完整。（本题共12分，每空3分）

```
#include<string.h>
```

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

```
class Person;
```

```
_____ (1) _____
```

```
{
```

```
public:
```

```
_____ (2) _____(char *p,char *q)
```

```
{
```

```
strcpy(club,p);
```

```
strcpy(name,q);
```

```
}
```

```
void visit(Person&);
```

```
private:
```

```
char club[24];
```

```
char name[24];
```

```
};
```

```
class Person
```

```
{
```

```
public:
```

```
Person(char *p,int y)
```

```
{
```

```
strcpy(name,p);
```

```

age=y;
}
_____ (3) _____

private:
char name[24];
int age;
};
_____ (4) _____
{
cout<<"姓名:"<<x.name<<endl;
cout<<"年龄:"<<x.age<<endl;
cout<<"可以使用的俱乐部:"<<club<<endl;
cout<<"介绍人:"<<name<<endl;
}
void main()
{
TennisClub a("明星俱乐部","李晓明");
Person b ("张静",28);
a.visit(b);
}

```

2、文件 file1.txt 中有10个字符，下面的程序将文件 file1.txt 的内容复制到文件 file2.txt 中。请将程序补充完整。（本题共12分，每空3分）

```

#include<iostream.h>
_____ (5) _____

#include<stdlib.h>

void main( )
{
fstream infile, outfile;
infile.open("file1.txt",ios::in);

```

```
if(!infile)
{
cout<<"file1.txt can't open.\n";
abort();
}
```

\_\_\_\_\_ (6) \_\_\_\_\_

```
if(!outfile)
{
cout<<"file2.txt can't open.\n";
abort();
}
```

```
char ch;
for(int i=0;i<10;i++)
```

```
{
infile>>ch;
cout<<ch;
_____ (7) _____
}
```

\_\_\_\_\_ (8) \_\_\_\_\_

```
outfile.close();
```

```
}
```

3、请将下面的程序补充完整，使得函数 **fun()** 具有多态性。（本题10分，(9)、(12)空每空3分，(10)、(11)空每空2分）

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

```
#include <math.h>
```

```
class Power
```

```
{
```

```
public:
```

```
Power(int i){x=i;}
```

\_\_\_\_\_ (9) \_\_\_\_\_

protected:

int x;

};

class Square:public Power

{

public:

\_\_\_\_\_ (10) \_\_\_\_\_

void display()

{

cout<<"x="<<x<<endl;

cout<<"x square="<<x\*x<<endl;

}

};

\_\_\_\_\_ (11) \_\_\_\_\_

{

public:

Cube(int i) : Power(i){}

void display() { cout << "x = " << x << endl;

cout<<"x cube="<<x\*x\*x<<endl;}

};

void fun(\_\_\_\_\_ (12) \_\_\_\_\_){p.display();}

void main()

{

Square squ(2);Cube cub(3);

fun(squ);fun(cub);

}