

# 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);

}

# 2008 级期末考试试卷（A 卷） 2009 年 6 月

## 一、单项选择(每空1分，共20分)

1. 类 aClass 的定义如下：

```
class aClass
{public:
    aClass(){}
    aClass(int i){value=new int(i);}
    int *value;
}
```

若要对类中的 value 赋值，则下面的语句中，正确的是（ ）。

A) aClass my; my.value=10;

B) aClass my; \*my.value=10;

C) aClass my; my.\*value=10;

D) aClass my(10);

2. 如果类 A 被说明成类 B 的友元，则（ ）。

A) 类 A 的成员函数不得访问类 B 的所有成员

B) 类 A 的成员函数可以访问类 B 的所有成员

C) 类 A 的成员即类 B 的成员

D) 类 B 的成员即类 A 的成员

3. 类对象占用内存的大小与（ ）有关。

A) 数据成员 B) 成员函数

C) 数据成员和成员函数 D) 私有成员

4. 假定 MyClass 为一类，执行 MyClass a,b(2),\*p[2];语句时会自动调用该类构造函数（ ）次。

A) 2 B) 3 C) 4 D) 5

5. 下列关于成员函数特征的描述中，错误的是（ ）。

A) 成员函数可以重载

B) 成员函数一定是内联函数

C) 成员函数可以设置参数的默认值

D) 成员函数可以是静态的

6. 下列关于析构函数的说法错误的是 ( )。

A) 析构函数有且仅有一个

B) 析构函数无任何函数类型

C) 析构函数和构造函数一样可以有参数

D) 析构函数的作用之一是在对象被撤销时收回先前分配的内存空间

7. 下列虚基类的声明中, 正确的是 ( )。

A) `class virtual B:public A`

B) `virtual class B:public A`

C) `class B: public A virtual`

D) `class B: virtual public A`

8. 基类的 ( ) 在私有继承时在派生类中为私有成员函数, 在公有和保护继承时在派生类中为保护成员函数。

A) 私有成员函数 B) 保护成员函数 C) 公有成员 D) 保护数据成员

9. 多继承的构造函数顺序可分为如下4步:

①所有非虚基类的构造函数按照它们被继承的顺序构造。

②所有虚基类的构造函数按照它们被继承的顺序构造。

③所有子对象的构造函数按照它们声明的顺序构造。

④派生类自己的构造函数体

这四个步骤的正确顺序是 ( )。

A) ④③①② B) ②④③① C) ②①③④ D) ③④①②

10. 在派生类中重新定义虚函数时, 除了 ( ), 其他方面都应基类中相应的虚函数保持一致。

A) 函数名 B) 参数 C) 函数体 D) 返回值类型

11. 关于纯虚函数和抽象类的描述中, 错误的是 ( )。

A) 纯虚函数是一种特殊的虚函数, 它没有具体的实现。

B) 抽象类是指具有纯虚函数的类。

C) 一个基类中说明有纯虚函数, 该类的派生类一定不再是抽象类。

D) 抽象类作为基类来使用, 其纯虚函数的实现由派生类给出。

12. 虚函数必须是类的 ( )。
- A) 友元函数 B) 成员函数 C) 构造函数 D) 析构函数
13. 友元运算符 `obj1>obj2` 被 C++ 编译器解释为 ( )。
- A) `operator>(obj1,obj2)` B) `>(obj1,obj2)`
- C) `obj1.operator>(obj2)` D) `obj2.operator>(obj1)`
14. 下列运算符中, ( ) 在 C++ 中不能重载。
- A) `?:` B) `+` C) `-` D) `<=`
15. 下列关于运算符重载正确的是 ( )。
- A) 运算符重载可以改变运算符的操作数个数
- 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) 运算符重载不可以改变语法结构
16. 使用 `fstream` 流类定义流对象并打开磁盘文件时, 文件的隐含打开方式是 ( )。
- A) `ios::in` B) `ios::out` C) `ios::in|ios::out` D) 没有默认, 必须指定打开方式
17. 下列语句不能用于打开 C 盘根目录下文件 `test.txt` 的语句是 ( )。
- A) `ifstream fin; fin.open("c:\\test.txt");`
- B) `ifstream fin("c:\\test.txt");`
- C) A 和 B
- D) `ifstream fin; fin("c:\\test.txt");`
18. 下列输出字符 'A' 的方法中, ( ) 是错误的。
- A) `cout<<put('A');` B) `cout<<'A';`
- C) `cout.put('A');` D) `char A='A'; cout<<A;`
19. 类模板的类型参数 ( )。
- A) 只可作为数据成员的类型
- B) 只可作为成员函数的返回类型
- C) 只可作为成员函数的参数类型
- D) 以上三者皆可
20. ( ) 是一个通用函数的描述, 它可以对不同类型的数据完成同一个任务。
- A) 函数模板 B) 重载函数 C) 递归函数 D) 模板函数

## 二、判断题（每空1分，共10分）

注意判断题结果（正确为 **T**，错误为 **F**）

1. 当某个对象调用类的成员函数时，该对象的地址自动传递给 **this** 指针。\_\_\_\_\_
2. 在一个类中声明了某个静态数据成员，并在类外给出其定义和初始化后，若未定义该类的对象，该静态数据成员则不存在。\_\_\_\_\_
3. 构造函数不能被派生类继承。\_\_\_\_\_
4. 派生类至少有一个基类。\_\_\_\_\_
5. C++语言类定义中默认访问权限是 **public**。\_\_\_\_\_
6. 对虚函数的调用，系统都是采用动态联编。\_\_\_\_\_
7. 假定要对类 **AB** 定义加号操作符重载成员函数，实现两个 **AB** 类对象的加法，并返回相加结果，则该成员函数的声明语句为：

**AB operator+(AB &,AB &);**\_\_\_\_\_

8. **cin** 是 **istream** 的一个对象，处理标准输入。\_\_\_\_\_
9. **getline()**函数只能用于从键盘上读取字符串。\_\_\_\_\_
10. **template <class T1,T2>**是正确的模板说明。\_\_\_\_\_

## 三、读程序写结果（每空2分，共38分）

1. 写出下面程序的运行结果。

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

```
class Sample
```

```
{ char ch;
```

```
public:
```

```
Sample() {ch='a';cout<<ch<<endl;}
```

```
Sample(char character) {ch=character;cout<<ch<<endl;}
```

```
~Sample()
```

```
{ cout<<"destruct"<<ch<<endl; }
```

```
};
```

```
void main()
```

```
{ Sample *first=new Sample;
```

```
Sample second('b') ,third('c');
```



```
delete first;
```

```
}
```

运行结果：

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

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

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

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

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

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

2. 写出下面程序的运行结果。

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

```
class Table
```

```
{
```

```
public:
```

```
Table(){cout<<"table"<<endl;}
```

```
~Table(){cout<<"table"<<endl;}
```

```
};
```

```
class Circle
```

```
{
```

```
public:
```

```
Circle(){cout<<"circle"<<endl;}
```

```
~Circle(){cout<<"circle"<<endl;}
```

```
};
```

```
class RoundTable:public Circle,public Table
```

```
{
```

```
public:
```

```
RoundTable():Table(),Circle()
```

```
{ cout<<"roundtable"<<endl; }
```

```
~RoundTable(){cout<<"roundtable"<<endl;}
```

```
};

void main()
{
    RoundTable rt;
}
```

运行结果：

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

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

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

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

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

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

3. 写出下面程序的运行结果。

```
#include<iostream.h>

class Instrument
{
public:
    virtual void play(){cout<<"play instrument"<<endl;}
    void display(){cout<<"instrument"<<endl;}
};

class Wind:public Instrument
{
public:
    void play(){cout<<"play wind"<<endl;}
    void display(){cout<<"wind"<<endl;}
};

class Stringed:public Instrument
{
public:
```

```

void play(){cout<<"play stringed"<<endl;}
void display(){cout<<"stringed"<<endl;}

};

void main()
{
    Wind w;
    Stringed s;
    Instrument *p=&w;
    p->play();
    w.play();
    p=&s;
    p->display();
    s.display();
    w.display();
}

```

运行结果：

```

(13) _____
(14) _____
(15) _____
(16) _____
(17) _____

```

4. 读程序，按要求写出结果。

```

#include<fstream.h>
#include<stdlib.h>

void main()
{
    ofstream outfile1("file1.txt");
    ofstream outfile2("file2.txt");
    if(!outfile1||!outfile2)

```

```

{
cout<<"file can't open"<<endl;
abort();
}

char str1[]="abcd";
outfile1<<str1;
char str2[]="efg";
outfile2<<str2;
outfile1.close();
outfile2.close();

ifstream infile("file2.txt");
ofstream outfile("file1.txt",ios::app);
char ch;
while(infile.get(ch))
outfile.put(ch);
infile.close();
outfile.close();
}

```

程序运行后，

file1.txt 文件内容为： (18) \_\_\_\_\_

file2.txt 文件内容为： (19) \_\_\_\_\_

#### 四、程序填空（每空2分，共32分）

1. 请将下面程序补充完整。

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

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

```
class Student
```

```
{public:
```

```
void (1)_____ ( (2) _____)
```

```
{ strcpy(name,s); score=x; sum=sum+score; }  
(3) _____ double GetSum(){return sum;}
```

private:

```
char name[20];
```

```
double score;
```

```
static double sum;
```

```
};
```

```
(4) _____; //静态数据成员初始化为0
```

```
void main()
```

```
{ Student s1,s2;
```

```
s1.Setdate("zhang",99);
```

```
s2.Setdate("li",88);
```

```
cout<<Student::GetSum()<<endl;
```

```
}
```

2. 下面程序输出为:

zhang

计算机

99

wang

请将程序补充完整。

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

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

```
class Person
```

```
{ char name[20];
```

```
public:
```

```
Person(char*n)
```

```
{ strcpy(name,n); }
```

```
void display()
```

```
{ cout<<name<<endl; }
```

```

};

class Teacher: (5)_____

{protected:

char course[20];

public:

Teacher(char *n,char*c):Person(n)

{ strcpy(course,c); }

};

class Student: (6)_____

{ protected:

double score;

public:

Student(char *n,double s):Person(n)

{ score=s; }

};

class GraduateStudent:public Student,public Teacher

{ char advisor[20];

public:

GraduateStudent(char *n,char*c,double s,char *a): (7)_____

{ strcpy(advisor,a); }

void display()

{

(8)_____ ;

cout<<course<<endl<<score<<endl<<advisor<<endl;

}

};

void main()

{ GraduateStudent gs("zhang","计算机",99,"wang");

gs.display ();

```

```
}
```

3. 请将下面程序补充完整。

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

```
class Point
```

```
{
```

```
double x,y;
```

```
public:
```

```
(9) _____ (double a,double b)
```

```
{
```

```
x=a;
```

```
y=b;
```

```
}
```

```
(10) _____ int operator==(Point&,Point&);
```

```
Point& operator+=(Point&);
```

```
void display()
```

```
{ cout<<x<<" "<<y<<endl; }
```

```
};
```

```
Point& (11) _____(Point& c)
```

```
{
```

```
x=x+c.x;
```

```
y=y+c.y;
```

```
return (12) _____ ;
```

```
}
```

```
int operator==(Point& a,Point& b)
```

```
{
```

```
if(a.x==b.x&& a.y==b.y)
```

```
return 1;
```

```
else
```

```
return 0;
```

```

}

void main()
{
    Point p1(3,4),p2(4,5);
    if(p1==p2)
        cout<<"相等"<<endl;
    else
        cout<<"不相等"<<endl;

    p1+=p2;
    p1.display();
    p2.display();
}

```

4. 下面程序的功能是：先将3条学生记录写入文件，然后查找姓名为李四的学生记录，从文件中将其读出并输出到屏幕上，请将程序补充完整。

```

#include<fstream.h>

#include<stdlib.h>

#include<string.h>

class Student
{
    char name[20];
    double score;
public:
    Student(char*n="",double s=0)
    { strcpy(name,n);score=s; }
    void input()
    { cin>>name>>score; }
    void output()
    { cout<<name<<endl<<score<<endl; }
    char* getname()

```



```
{ return name; }

};

void main()

{
(13)_____ iosfile;

iosfile.open("file.dat", ios::in|ios::out|ios::binary);

if(!iosfile)

{

cout<<" file.dat can't open"<<endl;

abort();

}

Student s[3]={Student("张三",99),Student("李四",88),Student("王五",77)};

iosfile.write((char *)s, (14)_____ );

iosfile.seekg(0, (15)_____ );

Student stu;

while(iosfile.read( (16)_____ ,sizeof(Student)))

{

if(strcmp(stu.getname(),"李四")==0)

stu.output();

}

iosfile.close();

}
```



# 2009 级期末考试试卷（A 卷） 2010 年 6 月

## 一、单项选择(每空1分，共20分)

1. 已知: `char *s="123456"`; 则 `cout<<s+2`; 的输出结果为 ( )。  
A) 123456 B) 3 C) 3456 D) 2
2. 已知: `char *s="abcde"`; 则 `sizeof(s)`、`sizeof(*s)`、`strlen(s)` 的值依次为 ( )。  
A) 6 1 5 B) 4 1 5 C) 6 5 4 D) 4 4 5
3. 下列语句中正确的是 ( )。  
A) `char *s; *s="abcdefg"`;  
B) `char *s; cin>>s`;  
C) `char *s1="abcd", *s2="efghijk"; strcpy(s1,s2)`;  
D) `char *s="abcdefg"; cout<<*s`;
4. C++ 中, 关于构造函数和析构函数, 正确的描述是 ( )。  
A) 在定义类时, 必须自定义构造函数和析构函数, 在创建对象时自动调用构造函数, 在释放对象时自动调用析构函数  
B) 构造函数和析构函数均可以重载  
C) 已知类 `Student` 以及 `Student *p`; 在使用 `p=new Student`; 时自动调用无参构造函数创建动态对象, 在 `delete p`; 时自动调用析构函数释放动态对象  
D) 构造函数和析构函数都可以成为虚函数
5. 关于拷贝构造函数的描述正确的是 ( )。  
A) 通常的拷贝构造函数的参数是对象的指针类型  
B) 如果不自定义拷贝构造函数, 系统提供默认的拷贝构造函数  
C) 如果有自定义的构造函数, 系统就不再提供拷贝构造函数  
D) 如果需要用已有对象为新创建的对象初始化时, 就必须自定义拷贝构造函数
6. 有关静态成员的描述错误的是 ( )。  
A) 某个类的静态数据成员由该类的所有对象所共享  
B) 类的公有静态数据成员既可以用类的对象访问, 也可以直接用作用域运算符“`::`”通过类名来访问  
C) 静态数据成员既可以是私有成员, 也可以是公有成员  
D) 类中一旦定义了静态数据成员, 就必须定义静态成员函数, 以便对静态数据成员进行操作

作

7. 一个类的友元函数或友元类能够通过成员访问运算符访问该类的 ( )。

A) 所有成员 B) 私有成员 C) 保护成员 D) 公有成员

8. 下面关于继承方式的描述中错误的是 ( )。

A) 公有继承时, 基类的公有成员和保护成员在派生类中都成为公有成员

B) 私有继承时, 基类的公有成员和保护成员在派生类中都成为私有成员

C) 保护继承时, 基类的公有成员和保护成员在派生类中都成为保护成员

D) 无论哪种继承方式, 基类中的私有成员在派生类中都无法直接访问

9. 类型兼容是指在基类对象可以出现的地方, 都可以使用公有派生类的对象, 已知:

```
class BaseClass
```

```
{ //...};
```

```
class DerivedClass:public BaseClass
```

```
{ //...};
```

```
BaseClass b,*pb;
```

```
DerivedClass d,*pd;
```

下面不属于类型兼容的是 ( )。

A) b=d;

B) BaseClass &bb=d;

C) pd=&b;

D) pb=&d;

10. 在派生类中重新定义虚函数时, 除了 ( ), 其他方面都必须与基类中相应的虚函数保持一致。

A) 参数个数 B) 参数类型 C) 函数名称 D) 函数体

11. 下列运算符中, 必须使用成员函数进行重载的是 ( )。

A) == B) = C) >> D) ++

12. 下列关于运算符重载的描述中, 错误的是 ( )。

A) 运算符重载不可以改变优先级

B) 运算符重载不可以改变结合性

C) 运算符重载不可以改变运算符的操作数个数

D) 加法运算符“+”和赋值运算符“=”都重载之后，意味着“+=”也被重载了

13. 有关运算符重载的说法错误的是（ ）。

A) 在一个类中，可以对一个操作符进行多次重载

B) 重载赋值运算符“=”时，为了保持原有特性，重载运算符函数中应该使用返回语句“return \*this;”

C) C++中所有的运算符都可以被重载

D) 如果在某个类中使用成员函数对运算符重载，其左操作数必须是该类的对象

14. 已知某个类的友元函数重载了+=和-，a，b，c是该类的对象，则“a+=b-c”被C++编译器解释为（ ）。

A) operator+=(a,operator-(b,c))

B) a.operator+=(b.operator-(c))

C) operator+=(a,b.operator-(c))

D) a.operator+=(operator-(b,c))

15. 下面4个选项中，专门用于读取单个字符的是（ ）。

A) cin.read() B) cin.get() C) cin.put() D) cin.getline()

16. 下列关于getline( )函数的叙述中，错误的是（ ）。

A) getline( )函数仅用于从键盘而不能从文件读取字符串

B) getline( )函数读取字符串长度是受限制的

C) getline( )函数读取字符串时，遇到终止符就停止

D) getline( )函数中所使用的终止符默认是换行符，也可指定其他终止符

17. 打开文件的方式中，（ ）以追加方式打开文件。

A) ios::in B) ios::out C) ios::app D) ios::trunc

18. 当使用ofstream流类定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为（ ）。

A) ios:: out| ios::binary

B) ios::in| ios::binary

C) ios::out

D) ios::in

19. 有关函数模板和模板函数说法错误的是（ ）。

A) 函数模板只是对函数的描述，编译器不为其产生任何执行代码，所以它不是一个实实在在

在的函数

B) 模板函数是实实在在的函数，它由编译系统在遇到具体函数调用时所生成，并调用执行

C) 函数模板需要实例化为模板函数后才能执行

D) 当函数模板和一般函数同名时，系统先去匹配函数模板，将其实例化后进行调用

20. 一个 ( ) 允许用户为类定义一种模式，使得类中的某些数据成员及某些成员函数的返回值能取任意类型。

A) 类模板 B) 模板类 C) 函数模板 D) 模板函数

## 二、判断题（每空1分，共20分）

注意：判断题结果（正确为 T，错误为 F）

1. 类定义中的成员默认访问权限是 `private`。\_\_\_\_\_

2. 一个类中的保护成员和公有成员类似，在程序的任何地方都可以被访问。\_\_\_\_\_

3. 系统提供的缺省构造函数没有参数，所以自定义构造函数必须带有参数。\_\_\_\_\_

4. 一旦自定义了构造函数，系统便不再提供缺省的构造函数。\_\_\_\_\_

5. 一个类只能有一个构造函数和一个析构函数。\_\_\_\_\_

6. 静态数据成员必须在类中进行定义和初始化。\_\_\_\_\_

7. 静态成员函数中不能访问非静态成员。\_\_\_\_\_

8. 重载插入运算符“<<”必须采用成员重载。\_\_\_\_\_

9. 如果类 A 是类 B 的友类，那么类 A 中的所有成员函数都可以访问类 B 中的所有成员。\_\_\_\_\_

10. 释放派生类的对象时，首先调用基类的析构函数，然后调用派生类的析构函数。\_\_\_\_\_

11. 拥有纯虚函数的类称为虚拟基类，它不能用来定义对象。\_\_\_\_\_

12. 虚函数只有在有继承的情况时才会存在。\_\_\_\_\_

13. 已知：`class Base1{//...};`

`class Base2{//...};`

`class Derived:public Base1,public Base2`

`{ Derived():Base2(),Base1(){} }`

`//...`

`};`

创建 `Derived` 类的对象时，先调用 `Base2` 的构造函数，然后调用 `Base1` 的构造函数，最后调用 `Derived` 的构造函数。\_\_\_\_\_

14. 基类的指针或引用调用虚函数时采用后期绑定。\_\_\_\_\_
15. 由抽象基类继承得到的派生类肯定是具体类。\_\_\_\_\_
16. 友元函数内能够访问任何对象的任何成员。\_\_\_\_\_
17. 对二元运算符采用成员函数重载时，只需要一个参数，而且该参数可以是内部类型。\_\_\_\_
18. 对一元运算符采用某个类的友元函数重载时需要一个参数，参数为该类的对象，不能是其他类型。\_\_\_\_\_
19. C++的输入/输出流库中，ios 类是一个虚基类，istream 类、ostream 类以及 streambuf 类都是 ios 类的派生类。\_\_\_\_\_
20. 设 inf 是一个 ifstream 类的流对象，则 inf.seekg(10,ios::beg);表示将文件指针从文件当前位置向后移动10个字节。\_\_\_\_\_

### 三、读程序写结果（每空2分，共32分）

1. 写出下面程序的运行结果。

```
#include<iostream.h>

class A
{ public:
    A( )
    { cout<<"A::A()called.\n"; }
    virtual ~A( )
    { cout<<"A::~~A()called.\n"; }
};

class B:public A
{ public:
    B(int i)
    { cout<<"B::B()called.\n";
      buffer=new char[i]; }
    virtual ~B( )
    { delete []buffer;
      cout<<"B::~~B()called.\n"; }
private:
```

```
char* buffer;  
  
};  
  
void fun(A* a)  
{ delete a; }  
  
void main()  
{ A *b=new B(10);  
  fun(b);  
}
```

运行结果:

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_
- (3) \_\_\_\_\_
- (4) \_\_\_\_\_

2. 写出下面程序的运行结果。

```
#include<iostream.h>  
  
class Shape  
{  
public:  
  void Draw()  
{  
  cout<<"Shape"<<endl;  
}  
  virtual void Print()=0;  
};  
  
class Circle:public Shape  
{  
private:  
  double r;  
public:
```



```
void Draw()
{
    cout<<"Circle"<<endl;
}

void SetData(double radius)
{
    r=radius;
}

void Print()
{
    cout<<"area:"<<3.14*r*r<<endl;
}

};

class Rectangle:public Shape
{
private:
    double a,b;
public:
    void Draw()
    {
        cout<<"Rectangle"<<endl;
    }

    void SetData(double x,double y)
    {
        a=x,b=y;
    }

    void Print()
    {
        cout<<"area:"<<a*b<<endl;
    }
}
```

```

}
};

void main()
{
    Circle c;
    Rectangle r;
    Shape *sp1=&c;
    Shape &sp2=r;
    c.SetData(10);
    r.SetData(3,5);
    sp1->Draw();
    c.Print();
    sp2.Draw();
    r.Print();
}

```

运行结果：

```

(5)_____
(6)_____
(7)_____
(8)_____

```

3. 写出下面程序的运行结果（注：运行结果中首行的空白行不考虑）。

```

#include<iostream.h>

class CArray
{ public:
    CArray(int i)
    { Length=i;
      Buffer=new char[Length+1]; }
    ~CArray()
    { delete []Buffer; }
}

```

```

int GetLength()
{ return Length; }

char& operator[](int i);

private:
int Length;
char* Buffer;

};

char& CArray::operator[](int i)
{ static char ch;
  if(i<Length&&i>=0)
    return Buffer[i];
  else
  { cout<<"\nIndex out of range.";
    return ch; }
}

void main()
{ int cnt;
  CArray string1(6);
  char *string2="Nankai";
  for(cnt=0;cnt<8;cnt++)
    string1[cnt]=string2[cnt];
  cout<<endl;
  for(cnt=0;cnt<8;cnt++)
    cout<<string1[cnt];
  cout<<"\n";
  cout<<string1.GetLength()<<endl;
}

```

运行结果：

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

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

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

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

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

(14) \_\_\_\_\_

4. 写出下面程序的运行结果。

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

```
void fun(char *s)
```

```
{
```

```
int n=0;
```

```
while(s[n]!='\0')
```

```
n++;
```

```
char t,*p=s+n-1;
```

```
while(s<p)
```

```
{
```

```
t=*s;
```

```
*s=*p;
```

```
*p=t;
```

```
s++;
```

```
p--;
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
char str[]="abcdefg";
```

```
fun(str);
```

```
cout<<str<<endl;
```

```
fun(str+1);
```

```
cout<<str<<endl;
```

}

运行结果:

(15) \_\_\_\_\_

(16) \_\_\_\_\_

**s**

#### 四、程序填空（每空2分，共28分）

1. 下面的程序是一个类模板，可实现求三个变量的和，请将程序补充完整。

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

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

```
class ff
```

```
{ Type a1, a2, a3;
```

```
public:
```

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

```
{ a1=b1; a2=b2; a3=b3; }
```

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

```
{ return a1+a2+a3; }
```

```
};
```

```
void main()
```

```
{ ff <int> x(12,13,14), y(16,17,18);
```

```
cout<<x.sum( )<<" "<<y.sum( )<<endl;
```

```
}
```

2. 下面的程序将一个普通函数作为类的友元函数，求坐标点之和，并且程序输出结果为2,2,4，请将程序补充完整。

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

```
class Point
```

```
{
```

```
int X,Y;
```

```
public:
```

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

```

{ X=x; Y=y; Countp++; }

Point(Point &p)

{ X=p.X; Y=p.Y; Countp++; }

~Point()

{ Countp--; }

(5)_____

static int Countp;

void display(){cout<<X<<" "<<Y<<" ";}

};

Point myfun(Point p1, Point p2, Point p3)

{ Point tmp(p1.X+p2.X+p3.X, p1.Y+p2.Y+p3.Y);

(6)_____

}

(7)_____

void main()

{ Point pp0,pp1(1,2),pp2(1);

Point p=myfun(pp0,pp1,pp2);

p.display ();

cout<< (8)_____ <<endl; // 输出 Countp 的值

}

```

3. 下面的程序将一个已有文件的内容复制到另一个文件中。请将程序补充完整。

```

#include<iostream.h>

#include<fstream.h>

#include<stdlib.h>

void main( )

{ (9)_____

infile.open("d:\\file1.txt",ios::in);

if(!infile)

{ cout<<"file1.txt can't open.\n";

```

```

abort(); }

outfile.open("d:\\file2.txt",ios::out);

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

char str[80]="\0";

while(!infile.eof())
{ infile.read(str,sizeof(str));
  (10)_____

  cout<<str;

}

cout<<endl;

infile.close();

(11)_____

}

```

4. ARRAY 类的定义如下，构造函数把参数 **n** 的值赋给 **s**，给 **v** 动态分配长度为 **n** 的数组空间，然后利用数组参数 **a** 初始化 **v** 所指向的数组。请将类定义补充完整。

```

class ARRAY

{ int *v;

  int s;

public:

  ARRAY( int a[], int n );

  ~ARRAY( )

  { delete []v; }

  int size()

  { return s; }

  int& operator[](int n);

};

(12)_____ ARRAY(int a[], int n)

```

```
{ if( n<=0 )  
{ v=NULL; s=0; return; }  
  
s=n;  
  
v= (13) _____  
  
for(int i=0; i<n; i++)  
(14) _____  
  
}
```



# 2007 级期末考试试卷（A 卷）答案

## 一、单项选择(每空1分，共20分)

CBDAD CDDAA CCBDD DBBAD

## 二、判断题(每空1分，共10分)

FTFTT FFFTT

## 三、读程序写结果(共36分)

(1) x=0,y=0 (2) x=2,y=3 (3) x!=y (4) x=y (5) n=10 (6) n=30 (7) s=5

(8) a=>-6 (9) s=0 (10) c=>T (11) Hello (12) 50 (13) OK!

## 四、程序填空（每空2至3分，共34分）

(1) class TennisClub

(2) TennisClub

(3) friend class TennisClub; 或 friend TennisClub; 分号可有可无

(4) void TennisClub::visit(Person &x)

(5) #include<fstream.h>

(6) outfile.open("file2.txt",ios::out);

(7) outfile<<ch;

(8) infile.close();

(9) virtual void display() {};

(10) Square(int i):Power(i) {}

(11) class Cube:public Power

(12) Power &p

# 2008 级期末考试试卷（A 卷）答案

## 一、单项选择（每空1分，共20分）

D B A A B C D B C C

C B A A D D D A D A

## 二、判断题（每空1分，共10分）

√ × √ √ × × × √ × ×

## 三、读程序写结果（每空2分，共38分）

(1) a (2) b (3) c (4) destruct a

(5) destruct c (6) destruct b (7) circle (8) table

(9) roundtable (10) roundtable (11) table (12) circle

(13) play wind (14) play wind (15) instrument (16) stringed

(17) wind (18) abcdefg (19) efg

## 四、程序填空（每空2分，共32分）

(1) Setdate (2) char \*s (or char s[]),double x (3) static (4) double Student::sum=0

(5) virtual public Person (6) virtual public Person

(7) Person(n),Teacher(n,c),Student(n,s) (8) Person::display ()

(9) Point (10) friend (11) Point::operator+= (12) \*this

(13) fstream (14) sizeof(s) or sizeof(Student)\*3 or....(15) ios::beg (16) (char\*)&stu

# 2009 级期末考试试卷（A 卷）答案

## 一、单项选择(每空1分，共20分)

1	2	3	4	5	6	7	8	9	10
<b>C</b>	<b>B</b>	<b>D</b>	<b>C</b>	<b>B</b>	<b>D</b>	<b>A</b>	<b>A</b>	<b>C</b>	<b>D</b>
11	12	13	14	15	16	17	18	19	20
<b>B</b>	<b>D</b>	<b>C</b>	<b>A</b>	<b>B</b>	<b>A</b>	<b>C</b>	<b>C</b>	<b>D</b>	<b>A</b>

## 二、判断题（每空1分，共20分）（正确为 T，错误为 F）

1	2	3	4	5	6	7	8	9	10
<b>T</b>	<b>F</b>	<b>F</b>	<b>T</b>	<b>F</b>	<b>F</b>	<b>T</b>	<b>F</b>	<b>T</b>	<b>F</b>
11	12	13	14	15	16	17	18	19	20
<b>F</b>	<b>T</b>	<b>F</b>	<b>T</b>	<b>F</b>	<b>F</b>	<b>T</b>	<b>T</b>	<b>F</b>	<b>F</b>

## 三、读程序写结果（每空2分，共32分）

- (1) **A::A()called.** (2) **B::B()called.** (3) **B::~~B()called.** (4) **A::~~A()called.**  
 (5) **Shape** (6) **area:314** (7) **Shape** (8) **area:15**  
 (9) **Index out of range.** (10) **Index out of range.** (11) **Nankai** (12) **Index out of range.**  
 (13) **Index out of range.** (14) **6** (15) **gfedcba** (16) **gabcdef**

## 四、程序填空（每空2分，共28分）

- (1) **template <class Type>** (2) **ff(Type b1, Type b2, Type b3)** (3) **Type sum()**  
 (4) **Point(int x=0, int y=0)** (5) **friend Point myfun(Point p1, Point p2, Point p3);**  
 (6) **return tmp;** (7) **int Point::Countp=0;** (8) **Point::Countp or pp0.Countp.....**  
 (9) **fstream infile, outfile;** (10) **outfile.write(str,sizeof(str));** (11) **outfile.close();**  
 (12) **ARRAY::** (13) **new int[n];** (14) **v[i]=a[i]; or \*(v+i)=\*(a+i);**

注：第1空 class 也可写为 typename；第5空中 p1 p2 p3 可缺省；第10空，如果写成 outfile.write(str,strlen(str)); 由于没有头文件 string.h，应该算错；但如果写成 outfile<<str<<endl; 应该是对的。