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 delelte 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

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的访问权限是 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)_____

⑽_____

⑾_____

⑿_____

⒀_____

⒁ _____

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>
⑴_____
class ff
{ Type a1, a2, a3;
public:
⑵_____
{ a1=b1; a2=b2; a3=b3; }
⑶_____
{ 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:
⑷_____
```

```
{ 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));
⑽_____
cout<<str;
}
cout<<endl;
infile.close();
⑾_____
}
```

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);
};
⑿_____ 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)_____
}
```