软件学院《C++与面向对象技术》期末考试样卷

(考试形式：开卷 考试时间：2小时)

注意：答案一定要写在答卷中，本试卷要和答卷一起交回，否则不给分。

一．单项选择题（20分）

1．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

2．下列关于模板(template)的说法正确的是

A）模板的实参在任何时候都可以省略
B）类模板与模板类所指的是同一概念
C）类模板的参数必须是类型
D）函数模板的类型形参理论上可以接受任意类型实参，但有可能对一些类型实参它不能正确运行

3．下面对类静态数据成员的描述中,正确的是

A）静态数据成员是类的所有对象共享的数据
B）类的每个对象都有自己的静态数据成员
C）类的不同对象有不同的静态数据成员值
D）静态数据成员不能通过类的对象调用

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）派生类的对象可以赋值给基类的对象
B）派生类的对象可以初始化基类的引用
C）派生类的对象的地址可以赋值给指向基类的指针
D）派生类的对象可以直接访问基类中的成员

5．下列关于多态性的描述，错误的是

A）C++语言中的多态性分为编译时的多态性和运行时的多态性
B）编译时的多态性可通过函数重载实现
C）运行时的多态性可通过模板和虚函数实现
D）实现运行时多态性的机制称为动态绑定

6．下列不是描述类的成员函数的是

A）构造函数
B）析构函数

C）友元函数
D）拷贝构造函数

7. 下面关于C++语言的描述错误的是
A）C++语言支持数据封装
B）静态数据成员可以在类体内进行初始化
C）C++语言允许函数和运算符重载
D）C++语言支持动态绑定(dynamic binding)

8. 对于类定义

```
class A{
public:
        virtual void func1（）{}
        void func2（）{}
};
class B:public A{
public:
        void func1（）{cout<<″class B func 1″ <<end1;}
        virtual void func2（）{cout<<″class B func 2″ <<end1;}
};
```

下面正确的叙述是
A） A::func2（）和B::func1（）都是虚函数
B） A::func2（）和B::func1（）都不是虚函数
C） B::func1（）是虚函数，而A::func2（）不是虚函数
D） B::func1（）不是虚函数，而A::func2（）是虚函数

9．要禁止修改指针变量p本身，又要禁止修改p所指向的数据，这样的指针应这样声明
A. const char *p="ABCD"; B. char const *p="ABCD";
C. char *const p="ABCD"; D. char const * const p="ABCD";

10. 下列程序中画线处应填入的语句是

```
class Base
{
public:
 void fun(){cout<<"Base of fun"<<endl;}
};
class Derived:public Base
{
 void fun()
 {
     _________  //调用基类的成员函数fun
     cout<<"Derived of fun"<<endl;
 }
};
```

A）fun（）；
B）Base.fun（）；
C）Base::fun（）；
D）Base->fun（）；

二．简答题(共 45 分)

1．explicit 和 protected 两个关键字在 C++中的作用。(4 分)

2．解释以下三条 C++语句的含义(semantics) （6 分）

```
vector<int> a(10);
vector<int> b[10];
vector<int> c(10,10);
```

3．下面是一个完整的类模板定义，请指出现其错误并改正。(3 分)

```
template <class T>
class Array
{
    T data;
public:
    Array(T t) { data = t; }
    void print();
};
void Array<T>::print()
{
    cout << data << endl;
}
```

4．试在【 】补充完成代码，把 vector v 中每个集合的每个元素顺序输出。（6 分）

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

int main()
{
    int A[12] = {1,2,3,4,5,6,7,8,9,10,11,12};
    set<int> s1(A,A+4),s2(A+4,A+8),s3(A+8,A+12);
    vector< set<int> > v;

    v.insert(v.begin(),s1);
    v.insert(v.begin(),s2);
```

```
        v.insert(v.begin(),s3);

        【  】

        return 0;
    }
```

5．下列程序在构造函数和析构函数中申请和释放类的数据成员int *a，申请时使用形参b初始化a，请填空(6分)。

```
class A
{
public:
    A（int b）;
    ~A（）;
private:
    int *a;
};
A::A（int b）
{
    【1】;
}
A::~A（）
{
    【2】;
}
```

6．写出下列程序的执行结果？(4分)

```
#include <iostream>
using namespace std;
class Sample
{
    int A;
    static int B;
public:
    Sample(int a){A=a,B+=a;}
    static void func(Sample s);
};

void Sample::func(Sample s)
{
    cout<<"A="<<s.A<<",B="<<B<<endl;
}

int Sample::B=0;
```

```
int main()
{
    Sample s1(2),s2(5);
    Sample::func(s1);
    Sample::func(s2);
    return 0;
}
```

7．写出下列程序的执行结果？(4 分)

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

template <class T>
class Sample
{
    T n;
public:
    Sample(T i){n=i;}
    void operator++(int);
    void disp(){cout<<"n="<<n<<endl;}
};
template <class T>
void Sample<T>::operator++(int)
{
    n+=1;
}
int main()
{
    Sample<char> s('a');
    s++;
    s.disp();
    return 0;
}
```

8．写出下列程序的执行结果？(4 分)

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

class Base
{
public:
    Base() { cout<<"Base()"<<endl;}
    ~Base() { cout<<"~Base()"<<endl;}
```

```
};
class Derived:public Base
{
public:
     Derived() { cout<<"Derived()"<<endl;}
     ~Derived() { cout<<"~Derived()"<<endl;}
};

int main(void)
{
     Derived *p = new Derived;
     delete p;
     return 0;
}
```

9．写出下列程序的执行结果？(4 分)

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

class Base
{
public:
     virtual void disp() { cout << "base class" << endl; }
};

class Derive1:public Base
{
public:
     void disp() { cout << "derive1 class" << endl; }
};

class Derive2:public Base
{
public:
     void disp() { cout << "derive2 class" << endl; }
};

int main(void)
{

     Base *p;
     Base b;
```

```
    Derive1 d1;
    Derive2 d2;

    p = &b;
    p->disp();

    p = &d1;
    p->disp();

    p = &d2;
    p->disp();

    return 0;
}
```

10．写出下面程序的运行结果为(4分)

```
#include<iostream>
using namespace std;
class A
{
    int num;
public:
    A(int i){num=i;}
    A(A &a){num=a.num++;}
    void print(){cout<<num;}
};
int main()
{
    A a(1);
    A b(a);
    a.print();
    b.print();

    return 0;
}
```

三．编程题(共 35 分)

1.(15 分) 设计一个点类 Point，包含横、纵两个坐标数据(必须是 private 数据成员)，由它派生出圆类 Circle(必须是 private 数据成员)，并添加一个半径数据，求其面积(定义的类能使得以下程序正确运行，分别输出圆

对象 c 的圆心(5,7),半径 9 和面积)。

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

int main()
{
    Circle c(5,7,9);

    cout << "圆心： (" << c.getx() << "," << c.gety() <<")" << endl;
    cout << "半径： " << c.getr() << endl;
    cout << "面积： " << c.area() << endl;

    return 0;
}
```

2. (20 分)试补充向量类 VECTOR 的定义，满足以下要求：
1)定义转换构造函数(从普通 double 类型一维数组转换成类对象)；
2)重载一元取负操作符-，完成向量取负运算(向量所有元素由正变负，由负变正)；
3)重载减法操作符-，完成向量减法运算；
4)重载前序自减操作符，使得--V 后，V.data 的每个元素值减 1(V 为 VECTOR 类型的对象，下同)；
5)重载后序自减操作符，使得 V--后，V.data 的每个元素值减 1。
(注意：以上所有操作符的重载都必须符合操作符原本的语义，比如--V 是“先减后使用”)

```
class VECTOR
{
    double data[10];
    //
};
```