继承与派生

1.概念填空题

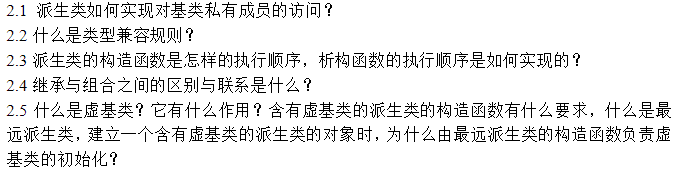
1.1在C++中，三种派生方式的说明符号为public、private、 protected不加说明，则默认的派生方式为private。

1.2当公有派生时，基类的公有成员成为派生类的公有成员；保护成员成为派生类的保护成员；私有成员成为派生类的不可见成员。当保护派生时，基类的公有成员成为派生类的保护成员；保护成员成为派生类的保护成员；私有成员成为派生类的不可见成员。

1.3 派生类的构造函数一般有3项工作要完成：首先初始化基类成员，其次初始化自身成员，最后执行函数体。

1.4多继承时，多个基类中的同名的成员在派生类中由于标识符不唯一而出现二义性。在派生类中采用虚基类或基类的作用域运算符来消除该问题。

1. 简答题



2.1

（1）将派生类声明为基类的友元

（2）将基类的私有成员设为保护成员

（3）在基类中定义一个公有属性的函数返回私有成员

2.2

1. 派生类的对象可以赋给基类的对象
2. 派生类的对象可以初始化基类对象的引用
3. 派生类对象的地址可以赋给指向基类对象的指针

2.3

首先初始化基类的成员，其次初始化自身的成员，最后执行函数体。

2.4

（1）继承是子类与父类之间的从属关系，组合是整体与部分的包含关系。

（2）继承与组合都是代码复用的方式。

2.5

1. 当在多条继承路径上有一个公共的基类,在这些路径中的某几条汇合处，这个公共的基类就会产生多个实例(或多个副本)，若只想保存这个基类的一个实例，可以将这个公共基类说明为虚基类
2. 避免二义性的产生
3. 含有虚基类的派生类的构造函数的初始值列表要调用虚基类的构造函数
4. 最远派生类是一个派生体系中最后派生出来的类
5. 可以保证虚基类的子对象只初始化一次

3．选择题

3.1下面对派生类的描述中，错误的是（D）。

A．一个派生类可以作为另外一个派生类的基类

B．派生类至少有一个基类

C．派生类的成员除了它自己的成员外，还包含了它的基类的成员

D．派生类中继承的基类成员的访问权限到派生类中保持不变

3.2下列对友元关系叙述正确的是（A）。

A．不能继承

B．是类与类的关系

C．是一个类的成员函数与另一个类的关系

D．提高程序的运行效率

3.3当保护继承时，基类的（B）在派生类中成为保护成员，不能通过派生类的对象来直接访问。

A．任何成员 B．公有成员和保护成员

C．公有成员和私有成员 D．私有成员

3.4设置虚基类的目的是（B）。

A．简化程序 B．消除二义性

C．提高运行效率 D．减少目标代码

3.5在公有派生情况下，有关派生类对象和基类对象的关系，不正确的叙述是（C）。

A．派生类的对象可以赋给基类的对象

B．派生类的对象可以初始化基类的引用

C．派生类的对象可以直接访问基类中的成员

D．派生类的对象的地址可以赋给指向基类的指针

3.6有如下类定义：

class MyBASE{

int k;

public:

void set(int n) {k=n;}

int get( ) const {return k;}

};

class MyDERIVED: protected MyBASE{

protected;

int j;

public:

void set(int m,int n){MyBASE::set(m);j=n;}

int get( ) const{return MyBASE::get( )+j;}

};

则类MyDERIVED中保护成员个数是（B）。

A．4 B．3 C．2 D．1

3.7程序如下：

#include<iostream>

using namespace std;

class A {

public:

A( ) {cout<<”A”;}

};

class B {public:B( ) {cout<<”B”;} };

class C: public A{

B b;

public:

C( ) {cout<<”C”;}

};

int main( ) {C obj; return 0;}

执行后的输出结果是（D）。

A．CBA B．BAC C．ACB D．ABC

3.8类O定义了私有函数F1。P和Q为O的派生类，定义为class P: protected O{…}； class Q: public O{…}。(C)可以访问Fl。

A． O的对象 B． P类内 C． O类内 D． Q类内

3.9有如下类定义：

class XA{

int x;

public:

XA(int n) {x=n;}

};

class XB: public XA{

int y;

public:

XB(int a,int b);

};

在构造函数XB的下列定义中，正确的是（B）。

A．XB::XB（int a，int b）：x(a)，y(b){ }

B．XB::XB（int a，int b）：XA(a)，y(b) { }

C．XB::XB（int a，int b）：x(a)，XB(b){ }

D．XB::XB（int a，int b）：XA(a)，XB(b){ }

4．写出程序运行结果

4.l#include<iostream>

using namespace std;

class B1{

public:

B1(int i){ cout<<”constructing B1 “<<i<<endl; }

~B1( ){ cout<<”destructing B1 “<<endl; }

};

class B2 {

public:

B2( ){ cout<<”constructing B3 \*”<<endl; }

~B2( ){ cout<<”destructing B3”<<endl; }

};

class C:public B2, virtual public B1 {

int j;

public:

C(int a,int b,int c):B1(a),memberB1(b) ,j(c){}

private:

B1 memberB1;

B2 memberB2;

};

int main( ){

C obj(1,2,3);

}

constructing B11

constructing B3 \*

constructing B12

constructing B3 \*

destructing B3

destructing B1

destructing B3

destructing B1