**1. 什么是虚基类，并简述其特点。**

**答：以virtual方式继承基类就是虚基类。将共同基类设置为虚基类时，从不同路经继承过来的同名数据成员只有一个副本，同一个函数也只有一个映射。**

**2. 派生类对基类成员的继承方式有哪些？**

**答：公有继承（public），私有继承（private），保护继承（protected）**

**3．C++的作用域有哪几种？**

**答：函数原型作用域，局部作用域，类作用域，文件作用域，命名空间作用域**

**4内联函数的实现?**

**答：内联函数不是在调用时发生控制转移，而是在编译时将函数体嵌入每一个调用处。这样就节省了参数传递、控制转移等开销。**

1. **什么是函数重载？**

**答：两个以上函数，具有相同的函数名，但是形参的个数或者类型不同，编译器根据实参和形参的类型及个数的最佳匹配，自动确定调用哪一个函数，这就是函数重载。**

1. **抽象是什么？**

**答：面向对象方法中的抽象，是指对具体问题进行概括，抽出一类对象的公共性质并加以描述的过程。对一个问题的抽象应该包括两个方面：数据抽象和行为抽象。**

1. **封装是什么？**

**答：封装就是将抽象得到的数据和行为（或功能）相结合，形成一个有机的整体，也就是将数据与操作数据的函数代码进行有机地结合，形成“类”，其中的数据和函数是类的成员。**

1. **多态性是什么？**

**答：多态性是指一段程序能够处理多种类型对象的能力。在C++语言中，这种多态性可以通过强制多态，重载多态，类型参数化多态，包含多态四种形式来实现。**

1. **类是什么？**

**答：类是对逻辑上相关的函数与数据封装，它是对问题的抽象描述。**

1. **构造函数的作用是什么？**

**答：构造函数的作用就是在对象被创建时利用特定的值构造对象，将对象初始化为一个特定的状态。构造函数在对象被创建的时候自动调用。**

1. **复制构造函数的作用？**

**答：复制构造函数的作用是使用一个已经存在的对象（由复制构造函数的参数指定），去初始化同类的一个新对象。**

**而隐含复制构造函数的功能是，把初始值对象的每个数据成员的值都复制到新建立的对象中。**

**12.复制构造函数的调用情况？**

**答：（1）当用类的一个对象去初始化该类的另一个对象时；**

**（2）如果函数的形参是类的对象时，调用函数，进行形参和实参结合时。**

**（3）如果函数的返回值是类的对象，函数执行完成返回调用者时。**

**13.析构函数功能？**

**答：析构函数在对象的生存期即将结束的时刻被自动调用的，它用来完成对象被删除前的一些清理工作。（析构函数不接受任何参数）**

1. **类的组合描述的是什么？**

**答：类的组合描述的就是一个类内嵌其他类的对象作为成员的情况，它们之间的关系是一种包含与被包含的关系。**

1. **UML的几种关系？**

**答：依赖关系、作用关系—关联、包含关系—聚集和组合、继承关系—泛化。**

**16.静态成员的功能？**

**答：解决同一个类的不同对象之间数据和函数共享问题。**

1. **什么是类属性？**

**答：类属性是描述类的所有对象共同特征的一个数据项，对于任何对象实例，它的属性值是相同的。**

1. **友元关系的功能？**

**答：友元关系提供了不同类或对象的成员函数之间、类的成员函数与一般函数之间进行数据共享的机制。**

1. **友元函数是什么？**

**答：友元函数是在类中用关键词friend修饰的非成员函数，友元函数可以是一个普通函数，也可以是其他类的成员函数，虽然它不是本类的成员函数，但是在它的函数体中可以通过对象名访问类的私有和保护成员。**

1. **友元类的解释？**

**答：若A类为B类的友元类，则A类的所有成员函数都是B类的友元函数，都可以访问B类的私有和保护成员。**

1. **友元关系的三个注意点？**

**答：（1）非传递 （2）单向性 （3）非继承**

1. **什么是数组？**

**答：数组是具有一定顺序关系的若干对象的集合体，组成数组的对象称为该数组的元素。**

1. **类的继承是什么？**

**答：类的继承是新的类从已有类那里得到已有的特性。**

1. **三个继承的特性？**

**答：**

|  |  |  |  |
| --- | --- | --- | --- |
| **继承方式** | **Public** | **Protected** | **Private** |
| **性质** | **基类公有和保护成员的访问属性在派生类中不变，而基类私有成员不可直接访问** | **基类公有和保护成员都以保护成员的身份出现在派生类中，而基类私有成员不可直接访问** | **基类公有和保护成员都以私有成员的身份出现在派生类中，而基类私有成员不可直接访问** |

1. **什么是类型兼容规则？**

**答：类型兼容规则是指在需要基类对象的任何地方，都可以使用公有派生类的对象来替代。**

**Class B{ ..... };**

**Class D:public B { ..... };**

**B b1,\*pb1;**

**D d1;**

**包括以下情况：①派生类的对象可以隐含转换为基类对象 b1=d1**

**②派生类的对象可以初始化基类对象的引用 B&rb=d1;**

**③派生类的指针可以隐含转换为基类的指针 pb1=&d1;**

1. **派生类的构造顺序是怎样的？**

**答：基类的构造函数 → 初始化列表中成员对象 → const & 基本类型成员 →派生类构造函数函数体**

1. **基类构造函数调用顺序是怎样的？**

**答：基类构造函数调用顺序是按照派生类定义时的顺序。**

**比如：class Derived：public Base 2，Base 1，Base 3{......}；**

**先Base2，再Base1，最后Base3**

**27.什么是作用域标识符可见性原则？**

**答：如果存在两个或多个具有包含关系的作用域，外层声明了一个标识符，而内层没有再次声明同名标识符，那么外层标识符再内层仍可见；如果在内层声明了同名标识符，则外层标识符在内层不可见，这时称内层标识符隐藏了外层同名标识符，这种现象称为隐藏规则。**

**28.函数重载、重写、同名覆盖是什么？**

**答：**

**函数重载：在作用域中函数名同名，函数参数类型个数不同时构成重载。**

**函数重写：在派生类中对基类成员函数的返回值类型、函数名、参数列表相同的函数重新定义，也叫虚函数重写。**

**函数覆盖：是指派生类的函数屏蔽了与其同名的基类函数。注意只要同名函数，不管参数列表是否相同，基类函数都会被隐藏。**

1. **虚基类的作用？**

**答：解决二义性问题。将共同基类设置为虚基类，这时从不同路径继承过来的同名数据成员在内存中就只有一个，同一个函数名也只有一个个映射。**

1. **静态绑定和动态绑定的解释？**

**答：绑定工作在编译连接阶段完成的情况称为静态绑定；绑定工作在程序运行阶段完成的情况称为动态绑定。**

1. **运算符重载是什么？**

**答：运算符重载是对已有的运算符赋予多重含义，使同一个运算符作用于不同类型的数据时导致不同的行为。**

1. **运算符重载规则？**

**答：①除个别运算符（如：类属关系运算符“.” 成员指针运算符“.\*” 作用域运算符“：：” 三目运算符“？：”）不能被重载，其余都可以重载，只能重载c++中已经存在的运算符。**

**②重载后运算符优先级和结合性都不变。**

**③一般来说，重载的功能与原有功能类似，不能改变运算符的操作对象个数，同时至少要有一个操作对象是自定义类型。**

1. **运行过程中的多态需要满足的条件？**

**答： ①类之间满足赋值兼容原则。**

**②声明虚函数。**

**③由成员函数来调用或者是通过指针、引用来访问虚函数。**

1. **纯虚函数：virtual 函数类型 函数名（参数表）= 0；**
2. **带有纯虚函数的类是抽象类。**
3. **抽象类不能实例化，即不能定义一个抽象类的对象。**