

C和C++

C语言是面向过程语言，重点在于算法和数据结构。C程序的设计首先考虑的是如何通过一个过程，对输入进行运算处理得到输出

C++语言是面向对象语言，首先考虑的是如何构造一个对象模型，让这个模型能够契合与之对应的问题，这样就可以通过获取对象的状态信息得到输出或实现过程控制。它在C的基础上添加了面向对象、模板等现在程序设计语言的特性。拓展了面向对象设计的内容，如类、继承、虚函数、模板和容器类等等，使之更加符合现代程序设计的需要。C++是C语言的继承，它既可以进行C语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计（泛型编程），还可以进行以继承和多态为特点的面向对象的程序设计（面向对象编程），支持类、封装、继承、多态等特性

面向对象编程：面向对象是一种对现实世界理解和抽象的方法、思想，通过将需求要素转化为对象进行问题处理的一种思想。

类：类定义了事物的属性和它的方法。一个类的属性和方法被称为“成员”。**类是在对象之上的抽象，对象则是类的具体化，是类的实例。**

封装性：封装使数据和加工该数据的方法（函数）封装为一个整体，把对象的设计者和对象的使用者分开，使用者不必知晓行为实现的细节，可以增加安全性。

继承性：继承性是子类共享父类之间数据和方法的机制。一个类直接继承其它类的全部描述，同时可修改和扩充。可以增强代码的复用性。

多态性：对象根据所接收的消息而做出动作。**同一消息为不同的对象接受时可产生完全不同的行动**，这种现象称为多态性。使具有不同内部结构的对象共享相同的外部接口。可以增加扩展性。

1、指针和变量的自增自减有什么不同？

答：变量的自增自减是改变变量的值，指针的自增自减是改变指针的指向地址。

2、宏定义与操作符的区别？

答：宏定义是C++的预处理命令之一，它是一个替换操作，不做计算和表达式求解，不占内存和编译时间。

3、虚函数与纯虚函数的特点？

答：虚函数必须是基类的非静态成员函数，其访问权限可以是protected或public；纯虚函数是虚函数的一个子集，含有纯虚函数的类就是抽象类，它不能生成对象。

虚函数：在类成员方法中声明 `virtual void func()`

纯虚函数：在类成员方法中生命 `virtual void func()=0`

对于虚函数，子类也可以定义也可以不重新定义基类的虚函数

对于纯虚函数，子类**必须重新定义**基类的纯虚函数

4、如何使用纯虚函数？

答：纯虚函数用来定义没有意义的实现，用于抽象类中需要交给派生类具体实现的方法。

5、引用与值传递的区别？

答：值传递传递的是一个值的副本，函数对形参的操作不会影响实参的值；引用传递传递的是引用对象的内存地址，函数对形参的操作会影响实参的值，实参的值会随着形参的值得改变而改变。

6、指针与引用的区别？

答：（1）引用无需解引用，指针需要解引用；（2）引用在定义时被初始化一次，之后不可变，指针可变；（3）引用不能为空，指针可以为空；（4）程序为指针变量分配内存区域，而引用不需要分配内存区域，所以指针自增操作是指针变量的自增，引用自增操作是变量值的自增。

7、面向对象与面向过程的区别？

答：面向过程是一种以过程为中心的编程思想，以算法进行驱动；面向对象是一种以对象为中心的编程思想，以消息进行驱动。面向过程编程语言的组成：程序=算法+数据；面向对象编程语言的组成：程序=对象+消息。

8、面向对象的特征是什么？

答：面对对象的3个要素：封装，继承，多态。面向对象中所有对象都可以归属为一个类。

9、类与结构体有什么区别？

答：（1）结构体存储在栈中，类的实例化可以存储在栈中，也可以存储在堆中；（2）结构体的执行效率比类要高；（3）结构体没有析构函数，类有析构函数；（4）结构体不可以继承，类可以继承。

10、C++覆盖与隐藏概述？

答：（1）覆盖指的是在子类和父类中，存在函数名、参数均相同的函数，并且父类的该函数为虚函数；（2）隐藏指的是在子类与父类中，存在函数名相同、参数不同的函数，此时无论父类函数是否为虚函数，父类函数都会被隐藏，或者存在函数名、参数均相同的函数，此时只有当父类函数不为虚函数时，父类函数才会被隐藏。

11、C++支持参数个数不确定的函数吗？

答：C++可以通过隐藏参数机制支持参数不确定的函数。

12、什么是内联函数？

答：在类声明的内部声明或定义的成员函数叫做内联（inline）函数，在内联函数内不允许有循环语句和switch语句。

13、什么是静态函数？如何使用静态函数？

答：静态函数是用static修饰符修饰的函数，静态函数没有this指针，只能访问静态变量。类中如果函数调用的结果不会访问或者修改任何对象数据成员，这样的成员声明为静态成员函数比较好。

14、函数重载及作用域？

答：函数重载是指在相同作用域下，具有相同名称而不同参数列表的多个函数。

15、什么是函数模板？

答：函数模板技术是指使用了模板技术定义了参数化类型的非成员函数，这使得程序能够使用不同的参数类型调用相同的函数。

16、什么是类模板？

答：类模板是使用模板技术的类，描述了能够管理其他数据类型的通用数据类型。类模板技术通常用于建立包含其他类型的容器类（队列、链表、堆栈等）。

17、什么是泛型编程？

答：泛型编程就是以独立于特定类实现的方式编写代码，针对不同的类型提供通用的实现。

18、C++如何实现泛型编程？

答：C++中泛型编程的实现是使用C++中的模板技术来实现的，主要是设计函数模板和类模板。

Python
