1、**继承和多态的区别**：

继承是指，子类可以自动获得父类提供的函数，如父类提供了foo函数，则子类也会有foo函数

多态是指，子类可以重写父类的某个函数，从而为这个函数提供不同于父类的行为。一个父类的多个子类可以为同一个函数提供不同的实现，从而在父类这个公共的接口下，表现出多种行为

2、**public、private和protected**：

public protected private 是访问级别，public的成员任何地方可以访问，protected的成员只有子类可以访问，private成员只有类自己的内部可以访问。但三者均可被友元或友元函数访问。

3. **指针和引用的区别**

1：引用是变量的一个别名，内部实现是只读指针

2：引用只能在初始化时被赋值，其他时候值不能被改变，指针的值可以在任何时候被改变

3：引用不能为NULL，指针可以为NULL

4：引用变量内存单元保存的是被引用变量的地址

5：“sizeof 引用" = 指向变量的大小 ， "sizeof 指针"= 指针本身的大小

6：引用可以取地址操作，返回的是被引用变量本身所在的内存单元地址

7：引用使用在源代码级相当于普通的变量一样使用，做函数参数时，内部传递的实际是变量地址

4、**const 有什么用途**

主要有三点：

1：定义只读变量，即常量

2：修饰函数的参数和函数的返回值

3： 修饰函数的定义体，这里的函数为类的成员函数，被const修饰的成员函数代表不修改成员变量的值

5、**堆和栈的区别**

一个由c/C++编译的程序占用的内存分为以下几个部分

1、栈区（stack）― 由编译器自动分配释放 ，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈。

2、堆区（heap） ― 一般由程序员分配释放， 若程序员不释放，程序结束时可能由OS回收 。

注意它与数据结构中的堆是两回事，分配方式倒是类似于链表，呵呵。

3、全局区（静态区）（static）―，全局变量和静态变量的存储是放在一块的，

初始化的全局变量和静态变量在一块区域， 未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。 - 程序结束后有系统释放

4、文字常量区 ―常量字符串就是放在这里的。 程序结束后由系统释放

5、程序代码区―存放函数体的二进制代码。

6、**头文件种的ifndef/define/endif 是干什么用的**

防止头文件被重复包含

7、**关键字static的作用**

1. 函数体内 static 变量的作用范围为该函数体，不同于 auto 变量， 该变量的内存只被分配一次，因此其值在下次调用时仍维持上次的值

2. 在模块内的 static 全局变量可以被模块内所有函数访问，但不能被模块外其他函数访问

3. 在模块内的static 函数只可被这一模块内的其他函数调用，这个函数的使用范围被限制在声明它的模块内

4. 在类的static 成员变量属于整个类所拥有，对类的所以对象只有一份拷贝

1. 在类中的 static 成员函数属于整个类所拥有，这个函数不接收 this 指针，因而只能访问类的 static 成员变量

8、 **在c++程序中调用被C编译器编译后的函数，为什么要加extern“C”**

C++语言支持函数重载，C语言不支持函数重载，函数被C++编译器编译后在库中的名字与C语言的不同，

假设某个函数原型为：

void foo(int x, inty);

该函数被C编译器编译后在库中的名字为: \_foo

而C++编译器则会产生像: \_foo\_int\_int 之类的名字。

为了解决此类名字匹配的问题，C++提供了C链接交换指定符号 extern "C"。

1. **进程同步方式**：互斥锁、条件变量、信号量
2. **常见算**法：堆排序、快速排序、归并排序
3. **静态链表和动态链表的差别**

12、 **什么时候要用虚析构函数**

通过基类的指针来删除派生类的对象时，基类的析构函数应该是虚的。否则其删除效果将无法实现。

一般情况下，这样的删除只能够删除基类对象，而不能删除子类对象，形成了删除一半形象，从而千万内存泄漏。

原因：

在公有继承中，基类对派生类及其对象的操作，只能影响到那些从基类继承下来的成员。

如果想要用基类对非继承成员进行操作，则要把基类的这个操作（函数）定义为虚函数。

那么，析构函数自然也应该如此：如果它想析构子类中的重新定义或新的成员及对象，当然也应该声明为虚的。

注意：

如果不需要基类对派生类及对象进行操作，则不能定义虚函数（包括虚析构函数），因为这样会增加内存开销。

13**、c++怎样让返回对象的函数不调用拷贝构造函数**

拷贝构造函数前加 “explicit” 关键字