OK，进入正题，下面就是《必须掌握的20道技术面试题》。

**问1：请用简单的语言告诉我C++ 是什么？**

答：C++是在C语言的基础上开发的一种面向对象编程语言，应用广泛。C++支持多种编程范式 －－面向对象编程、泛型编程和过程化编程。 其编程领域众广，常用于系统开发，引擎开发等应用领域，是最受广大程序员受用的最强大编程语言之一,支持类：类、封装、重载等特性!

**问2：C和C++的区别？**

答：c++在c的基础上增添类，C是一个结构化语言，它的重点在于算法和数据结构。C程序的设计首要考虑的是如何通过一个过程，对输入（或环境条件）进行运算处理得到输出（或实现过程（事务）控制），而对于C++，首要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域，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（事务）控制。

**问3：什么是面向对象（OOP）？**

答：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。

**问4：什么是多态？**

答：多态是指相同的操作或函数、过程可作用于多种类型的对象上并获得不同的结果。不同的对象，收到同一消息可以产生不同的结果，这种现象称为多态。

**问5：设计模式懂嘛，简单举个例子？**

答：设计模式（Design pattern）是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。

比如单例模式，保证一个类仅有一个实例，并提供一个访问它的全局访问点。

适用于：当类只能有一个实例而且客户可以从一个众所周知的访问点访问它时；当这个唯一实例应该是通过子类化可扩展的，并且客户应该无需更改代码就能使用一个扩展的实例时。

比如工厂模式，定义一个用于创建对象的接口，让子类决定实例化哪一个类。Factory Method 使一个类的实例化延迟到其子类。

适用于：当一个类不知道它所必须创建的对象的类的时候；当一个类希望由它的子类来指定它所创建的对象的时候；当类将创建对象的职责委托给多个帮助子类中的某一个，并且你希望将哪一个帮助子类是代理者这一信息局部化的时候。

**问6：STL库用过吗？常见的STL容器有哪些？算法用过哪几个？**

答：STL包括两部分内容：容器和算法。（重要的还有融合这二者的迭代器）

容器，即存放数据的地方。比如array等。

在STL中，容器分为两类：序列式容器和关联式容器。

序列式容器，其中的元素不一定有序，但都可以被排序。如：vector、list、deque、stack、queue、heap、priority\_queue、slist；

关联式容器，内部结构基本上是一颗平衡二叉树。所谓关联，指每个元素都有一个键值和一个实值，元素按照一定的规则存放。如：RB-tree、set、map、multiset、multimap、hashtable、hash\_set、hash\_map、hash\_multiset、hash\_multimap。

下面各选取一个作为说明。

vector：它是一个动态分配存储空间的容器。区别于c++中的array，array分配的空间是静态的，分配之后不能被改变，而vector会自动重分配（扩展）空间。

set：其内部元素会根据元素的键值自动被排序。区别于map，它的键值就是实值，而map可以同时拥有不同的键值和实值。

算法，如排序，复制……以及个容器特定的算法。这点不用过多介绍，主要看下面迭代器的内容。

迭代器是STL的精髓，我们这样描述它：迭代器提供了一种方法，使它能够按照顺序访问某个容器所含的各个元素，但无需暴露该容器的内部结构。它将容器和算法分开，好让这二者独立设计。

**问7：数据结构会吗？项目开发过程中主要用到那些？**

答：数据结构中主要会用到数组，链表，树（较少），也会用到栈和队列的思想。

**问8：const知道吗？解释其作用。**

答：

1.const 修饰类的成员变量，表示成员常量，不能被修改。

2.const修饰函数承诺在本函数内部不会修改类内的数据成员，不会调用其它非 const 成员函数。

3.如果 const 构成函数重载，const 对象只能调用 const 函数，非 const 对象优先调用非 const 函数。

4.const 函数只能调用 const 函数。非 const 函数可以调用 const 函数。

5.类体外定义的 const 成员函数，在定义和声明处都需要 const 修饰符。。

**问9：类的static变量在什么时候初始化？函数的static变量在什么时候初始化？**

答：类的静态成员变量在类实例化之前就已经存在了，并且分配了内存。函数的static变量在执行此函数时进行初始化。

**问10：堆和栈的区别？堆和栈的生命周期？**

答：

**一、堆栈空间分配区别：**

1、栈（操作系统）：由操作系统自动分配释放 ，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（操作系统）： 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，分配方式倒是类似于链表。

**二、堆栈缓存方式区别：**

1、栈使用的是一级缓存， 他们通常都是被调用时处于存储空间中，调用完毕立即释放；

2、堆是存放在二级缓存中，生命周期由虚拟机的垃圾回收算法来决定（并不是一旦成为孤儿对象就能被回收）。所以调用这些对象的速度要相对来得低一些。

**三、堆栈数据结构区别：**

堆（数据结构）：堆可以被看成是一棵树，如：堆排序；

栈（数据结构）：一种先进后出的数据结构。

**问11：C和C++的区别？**

答：C++在C的基础上增添类

C是一个结构化语言，它的重点在于算法和数据结构。

C程序的设计首要考虑的是如何通过一个过程，对输入（或环境条件）进行运算处理得到输出（或实现过程（事务）控制），而对于C++，首要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域，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（事务）控制。

**问12：解释下封装、继承和多态？**

答：

**一、封装：**

封装是实现面向对象程序设计的第一步，封装就是将数据或函数等集合在一个个的单元中（我们称之为类）。

封装的意义在于保护或者防止代码（数据）被我们无意中破坏。

**二、继承：**

继承主要实现重用代码，节省开发时间。

子类可以继承父类的一些东西。

**三、多态**

多态：同一操作作用于不同的对象，可以有不同的解释，产生不同的执行结果。在运行时，可以通过指向基类的指针，来调用实现派生类中的方法。

**问13：指针和引用的区别？**

答：

1. 指针是一个变量，只不过这个变量存储的是一个地址，指向内存的一个存储单元；而引用仅是个别名；

2. 引用使用时无需解引用(\*)，指针需要解引用；

3. 引用只能在定义时被初始化一次，之后不可变；指针可变；

4. 引用没有 const，指针有 const；

5. 引用不能为空，指针可以为空；

6. “sizeof 引用”得到的是所指向的变量(对象)的大小，而“sizeof 指针”得到的是指针本身的大小；

7. 指针和引用的自增(++)运算意义不一样；

8. 指针可以有多级，但是引用只能是一级（int \*\*p；合法 而 int &&a是不合法的）

9.从内存分配上看：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。

**问14：什么是内存泄漏？面对内存泄漏和指针越界，你有哪些方法？你通常采用哪些方法来避免和减少这类错误？**

答：用动态存储分配函数动态开辟的空间，在使用完毕后未释放，结果导致一直占据该内存单元即为内存泄露。

使用的时候要记得指针的长度。

malloc的时候得确定在那里free.

对指针赋值的时候应该注意被赋值指针需要不需要释放.

动态分配内存的指针最好不要再次赋值.

**问15：常用的排序算法有哪些？简单描述几个排序算法的优缺点？**

答：选择、冒泡、快速、\*\*、希尔、归并、堆排等。

1.快排：是冒泡排序的一种改进。

优点：快，数据移动少

缺点：稳定性不足

2.归并：分治法排序，稳定的排序算法，一般用于对总体无序，但局部有序的数列。

优点：效率高O(n)，稳定

缺点：比较占用内存

**问16：new和malloc的区别？**

答：

1、malloc与free是C++/C语言的标准库函数，new/delete是C++的运算符。它们都可用于申请动态内存和释放内存。

2、对于非内部数据类型的对象而言，光用maloc/free无法满足动态对象的要求。对象在创建的同时要自动执行构造函数，对象在消亡之前要自动执行析构函数。

3、由于malloc/free是库函数而不是运算符，不在编译器控制权限之内，不能够把执行构造函数和析构函数的任务强加于malloc/free。因此C++语言需要一个能完成动态内存分配和初始化工作的运算符new，以一个能完成清理与释放内存工作的运算符delete。注意new/delete不是库函数。

4、C++程序经常要调用C函数，而C程序只能用malloc/free管理动态内存。

5、new可以认为是malloc加构造函数的执行。new出来的指针是直接带类型信息的。而malloc返回的都是void指针。

**问17：TCP和UDP通信的差别？什么是IOCP？**

答：

1.TCP面向连接， UDP面向无连接的

2.TCP有保障的，UDP传输无保障的

3.TCP是效率低的，UDP效率高的

4.TCP是基于流的，UDP基于数据报文

5.TCP传输重要数据，UDP传输不重要的数据

IOCP全称I/O Completion Port，中文译为I/O完成端口。

IOCP是一个异步I/O的API，它可以高效地将I/O事件通知给应用程序。

与使用select()或是其它异步方法不同的是，一个套接字[socket]与一个完成端口关联了起来，然后就可继续进行正常的Winsock操作了。然而，当一个事件发生的时候，此完成端口就将被操作系统加入一个队列中。然后应用程序可以对核心层进行查询以得到此完成端口。

**问18：同步IO和异步IO的区别？**

答：

**A. 同步**

所谓同步，就是在发出一个功能调用时，在没有得到结果之前，该调用就不返回。

按照这个定义，其实绝大多数函数都是同步调用（例如sin isdigit等）。

但是一般而言，我们在说同步、异步的时候，特指那些需要其他部件协作或者需要一定时间完成的任务。

最常见的例子就是 SendMessage。

该函数发送一个消息给某个窗口，在对方处理完消息之前，这个函数不返回。

当对方处理完毕以后，该函数才把消息处理函数所返回的值返回给调用者。

**B. 异步**

异步的概念和同步相对。

当一个异步过程调用发出后，调用者不会立刻得到结果。

实际处理这个调用的部件是在调用发出后，通过状态、通知来通知调用者，或通过回调函数处理这个调用。

**问19：解释C++中静态函数和静态变量？**

答：

(1)类静态数据成员在编译时创建并初始化：在该类的任何对象建立之前就存在，不属于任何对象，而非静态类成员变量则是属于对象所有的。类静态数据成员只有一个拷贝，为所有此类的对象所共享。

(2)类静态成员函数属于整个类，不属于某个对象，由该类所有对象共享。

1、static 成员变量实现了同类对象间信息共享。

2、static 成员类外存储，求类大小，并不包含在内。

3、static 成员是命名空间属于类的全局变量，存储在 data 区的rw段。

4、static 成员只能类外初始化。

5、可以通过类名访问（无对象生成时亦可），也可以通过对象访问。

1、静态成员函数的意义，不在于信息共享，数据沟通，而在于管理静态数据成员，完成对静态数据成员的封装。

2、静态成员函数只能访问静态数据成员。原因：非静态成员函数，在调用时 this指针时被当作参数传进。而静态成员函数属于类，而不属于对象，没有 this 指针。

**问20：说下你对内存的了解？**

答：

1.栈 - 由编译器自动分配释放

2.堆 - 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

3.全局区(静态区)，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。- 程序结束释放

4.另外还有一个专门放常量的地方。- 程序结束释放

5 程序代码区，存放2进制代码。

在函数体中定义的变量通常是在栈上，用malloc， calloc， realloc等分配内存的函数分配得到的就是在堆上。在所有函数体外定义的是全局量，加了static修饰符后不管在哪里都存放在全局区(静态区)，在所有函数体外定义的static变量表示在该文件中有效，不能extern到别的文件用，在函数体内定义的static表示只在该函数体内有效。另外，函数中的"adgfdf"这样的字符串存放在常量区。