1、extern的作用是什么？

用于提供一个全局变量的引用，全局变量对所有的程序文件都是可见的。可以这么理解，extern是用来在另一个文件中声明一个全局变量或函数。

2、static的作用是什么？

static的作用：常用来修饰变量。

使用 static 修饰局部变量可以在函数调用之间保持局部变量的值，而不需要在每次它进入和离开作用域时进行创建和销毁。

3、const的作用是什么？

const关键字可以用于修饰变量，参数，返回值，甚至函数体。const可以提高程序的健壮性，减少程序出错。它所限定的变量是不允许被改变的，从而起到保护的作用

4、指针与引用的区别是什么？

答：

①引用必须被初始化，指针不必。

②引用初始化以后不能被改变，指针可以改变所指的对象。

③不存在指向空值的引用，但是存在指向空值的指针。

5、new与malloc的区别是什么？

C语言提供了malloc和free函数，完成对堆内存的申请和释放。而C++则提供了两个关键字new和delete；

主要区别：

1.使用new操作符申请内存分配时无须指定内存块的大小，编译器会根据类型信息自行计算；malloc则需要显式地指出所需内存的尺寸。

2.new和delete是关键字，需要编译器支持；malloc和free是库函数，需要头文件支持；

3.new操作符内存分配成功时，返回的是对象类型的指针；而malloc内存分配成功则是返回void\*，因此，malloc的返回值一般都需要进行类型转换。

4.new如果分配失败了就会抛出bad\_malloc的异常，而malloc失败了则会返回NULL。

5.C++允许重载new/delete操作符，malloc不允许重载。

6、C++的多态性指什么？

多态定义：对于不同对象接收相同消息时产生不同的动作。

C++的多态性具体体现在运行和编译两个方面：在程序运行时的多态性通过继承和虚函数来体现；在程序编译时多态性体现在函数和运算符的重载上。

7、内联函数，宏定义和普通函数的区别？

一、宏定义和内联函数的区别

1、 宏定义不是函数，但是使用起来像函数。预处理器用复制宏代码的方式代替函数的调用，省去了函数压栈退栈过程，提高了效率。

内联函数本质上是一个函数，内联函数一般用于函数体的代码比较简单的函数，不能包含复杂的控制语句，while、switch，并且内联函数本身不能直接调用自身。如果内联函数的函数体过大，编译器会自动的把这个内联函数变成普通函数。

2、宏定义是在预编译的时候把所有的宏名用宏体来替换，简单的说就是字符串替换

内联函数则是在编译的时候进行代码插入，编译器会在每处调用内联函数的地方直接把内联函数的内容展开，这样可以省去函数的调用的开销，提高效率

3、 宏定义是没有类型检查的，无论对还是错都是直接替换。

内联函数在编译的时候会进行类型的检查，内联函数满足函数的性质，比如有返回值、参数列表等。

4、宏定义和内联函数使用的时候都是进行代码展开。不同的是宏定义是在预编译的时候把所有的宏名替换，内联函数则是在编译阶段把所有调用内联函数的地方把内联函数插入。这样可以省去函数压栈退栈，提高了效率

二、内联函数和普通函数的区别

内联函数和普通函数的参数传递机制相同，但是编译器会在每处调用内联函数的地方将内联函数内容展开，这样既避免了函数调用的开销又没有宏机制的缺陷

普通函数在被调用的时候，系统首先要到函数的入口地址去执行函数体，执行完成之后再回到函数调用的地方继续执行，函数始终只有一个复制。

内联函数不需要寻址，当执行到内联函数的时候，将此函数展开，如果程序中有N次调用了内联函数则会有N次展开函数代码

内联函数有一定的限制，内联函数体要求代码简单，不能包含复杂的结构控制语句。如果内联函数函数体过于复杂，编译器将自动把内联函数当成普通函数来执行

8、数组指针和指针数组的区别

数组指针只是一个指针变量，似乎是C语言里专门用来指向二维数组的，它占有内存中一个指针的存储空间。

指针数组是多个指针变量，以数组形式存在内存当中，占有多个指针的存储空间。

9、C++中类与结构体的区别？

1、结构体基本用在很多数据，需要一个结构体来封装这些数据。而类的话，是面向对象的思想，可以有很多接口让人调用，私有变量等外部不能调用，还有保护类型的变量。

2、结构体也可以被认为是一种特殊的类，它不存在任何函数，构造和析构函数也没有，而且是一个公共的的类。

3、结构体在默认情况下成员是公共的（public），类在默认情况下成员是私有的(private)。

4、C++结构体内部成员变量及成员函数默认的访问级别是public，而c++类的内部成员变量及成员函数的默认访问级别是private。

5、C++结构体的继承默认是public，而c++类的继承默认是private。

6、类要加上public变成共有的才能被访问，而结构本身就是共有的可直接访问。

7、C定义结构体变量时需要加struct关键字，C++中定义结构体变量时可以不加struct关键字。

10、析构函数的作用？

作用：对象消亡时，自动被调用，用来释放对象占用的空间。

11、虚函数的作用？

虚函数的作用是允许在派生类中重新定义与基类同名的函数，并且可以通过基类指针或引用来访问基类和派生类中的同名函数。

12、c语言和c++有什么区别？

1、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）。

2、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。 所以C与C++的最大区别在于它们的用于解决问题的思想方法不一样。之所以说C++比C更先进，是因为“ 设计这个概念已经被融入到C++之中 ”。

13、内存溢出，内存泄漏的原因？

1.内存中加载的数据量过于庞大，如一次从数据库取出过多数据；

2.集合类中有对对象的引用，使用完后未清空，使得JVM不能回收；

3.代码中存在死循环或循环产生过多重复的对象实体；

4.使用的第三方软件中的BUG；

5.启动参数内存值设定的过小

14、函数参数传递中值传递、地址传递、引用传递有什么区别？

1.值传递：形参拷贝实参的值，有内存分配和释放；不会改变原始值。

2.地址传递：形参为指针变量，指针的值为拷贝实参指针的值，有内存分配和释放；会改变原始值。

3.引用传递：形参是实参的引用，没有内存分配；会改变原始值。

15、链表和数组的区别在哪里？

数组：数据顺序存储，固定大小；

链表：数据可以随机存储，大小可动态改变。

16、C++重写、重载、重定义的区别？

\*\*1、重载overload：\*\*是函数名相同，参数列表不同 重载只是在类的内部存在。但是不能靠返回类型来判断。

\*\*2、重写override：\*\*也叫做覆盖。子类重新定义父类中有相同名称和参数的虚函数。函数特征相同。但是具体实现不同，主要是在继承关系中出现的 。

重写需要注意：

1 被重写的函数不能是static的。必须是virtual的

2 重写函数必须有相同的类型，名称和参数列表

3 重写函数的访问修饰符可以不同。尽管virtual是private的，派生类中重写改写为public,protected也是可以的

3、重定义 (redefining)也叫做隐藏:

子类重新定义父类中有相同名称的非虚函数 ( 参数列表可以不同 ) 。

17、请说出STL标准模板库中的几个常用类？

vector（向量）

set（集合）

string（字符串）

map（地图）

queue（队列）

stack（栈）

pair（对）

18、面向对象的三大特征是什么？

四大特性（重点考察点）

封装

继承

多态

抽象

（问三大特性，回答前三个即可）

19、构造函数与析构函数的异同点

构造函数是一个特殊的成员函数，能够自动初始化对象。

析构函数是在一个销毁对象时自动调用的成员函数。

通俗地讲，构造函数就是在对象刚诞生时执行的函数，析构函数则是在对象将被破坏（消亡）前执行的函数。

20、简述C++异常处理方式

在C++中,异常的抛出和处理主要使用了以下三个关键字:try、 throw 、 catch。

21、成员函数和友元函数的区别？

相同点：

1.对类的存取方式相同，可以直接存取类的任何存取控制属性的成员

2.可以通过对象存取形参、函数体中该类类型对象的所有成员

不同点：

1. 成员函数有this指针，友元函数没有。

2.友元函数不能被继承。

22、局部变量能否和全局变量重名？

能，局部会屏蔽全局。要用全局变量，需要使用"::"；

局部变量可以与全局变量同名，在函数内引用这个变量时，会用到同名的局部变量，而不会用到全局变量；

对于有些编译器而言，在同一个函数内可以定义多个同名的局部变量，如：在两个循环体内都定义一个同名的局部变量，而那个局部变量的作用域就在那个循环体内。

23、全局变量和局部变量在内存中是否有区别？如果有，是什么区别？

全局变量储存在静态数据库，局部变量在堆栈。

24、堆栈溢出一般是由什么原因导致的？

没有回收垃圾资源。

25、如何引用一个已经定义过的全局变量？

可以用引用头文件的方式，也可以用extern关键字。

如果用引用头文件方式来引用某个在头文件中声明的全局变量，假定你将那个变写错了，那么在编译期间会报错；

如果你用extern方式引用时，假定你犯了同样的错误，那么在编译期间不会报错，而在连接期间报错。

26、Heap与stack的差别？

Heap是堆，stack是栈。

Stack的空间由操作系统自动分配/释放，Heap的空间手动分配/释放;

Stack空间有限，Heap是很大的自由存储区;

C中的malloc函数分配的内存空间即在堆上,C++中对应的是new操作符。

程序在编译期对变量和函数分配内存都在栈上进行,且程序运行过程中函数调用时参数的传递也在栈上进行。

27、指针和引用有何区别？（主要区别）

初始化

引用必须在创建时被初始化；

指针可以在任何时间被初始化；

是否可变

引用只能在定义时被初始化一次（从一而终），之后不可变；

指针可变;

是否可为空

引用不能为空，引用必须连接到一块合法的内存；

指针可以为空。

this指针

C++中每一个对象都能通过this指针来访问自己的地址。this指针是所有成员函数的隐含参数。

使用：在成员函数内部，它可以用来指向调用对象。

注意：友元函数没有this指针，因为友元不是类的成员。只有成员函数才有 this 指针。

**JVM、JRE及JDK的关系** **＊＊**

JDK（Java Development Kit）是针对Java开发员的产品，是整个Java的核心，包括了Java运行环境JRE、Java工具和Java基础类库。

Java Runtime Environment（JRE）是运行JAVA程序所必须的环境的集合，包含JVM标准实现及Java核心类库。

JVM是Java Virtual Machine（Java虚拟机）的缩写，是整个java实现跨平台的最核心的部分，能够运行以Java语言写作的软件程序。

**简单来说就是JDK是Java的开发工具，JRE是Java程序运行所需的环境，JVM是Java虚拟机．它们之间的关系是JDK包含JRE和JVM，JRE包含JVM．**

**JAVA语言特点　＊＊**

* Java是一种面向对象的语言
* Java通过Java虚拟机实现了平台无关性，一次编译，到处运行
* 支持多线程
* 支持网络编程
* 具有较高的安全性和可靠性

**JAVA和C++的区别　＊＊**

面试时记住前四个就行了

* Java 通过虚拟机从而实现跨平台特性，但是 C++ 依赖于特定的平台。
* Java 没有指针，它的引用可以理解为安全指针，而 C++ 具有和 C 一样的指针。
* Java 支持自动垃圾回收，而 C++ 需要手动回收。
* Java 不支持多重继承，只能通过实现多个接口来达到相同目的，而 C++ 支持多重继承。
* Java 不支持操作符重载，虽然可以对两个 String 对象执行加法运算，但是这是语言内置支持的操作，不属于操 作符重载，而 C++ 可以。
* Java 的 goto 是保留字，但是不可用，C++ 可以使用 goto。