基本原理：设计这个特性是为了帮助解决哪类问题？其设计原理是什么？它有什么根本的局限？

规范：它该如何定义？

例子：当单独使用这个特性或与其他特性组合使用时，如何用好它？其中的关键技术和习惯用法是怎样的？在程序的可维护性和性能方面是否有一些隐含的问题？

C++

基础

C++和 C 区别

`struct` 与 `class` 区别

面向对象的三大特性：封装，继承，多态。面向对象的理解

类的访问权限：`private, protected, public`

STL

介绍下 STL（`vector, map` 等，内存分配等角度）

迭代器，空间配置器理解

`map` 如何实现的

`map` 里如何正确删除元素（注意 `iterator` 失效问题）

`list` 和 `vector` 实现的区别

`map vs unordered_map`

`set vs unordered_set`

仿函数，相对于普通函数的优点

`remove() vs erase()`

STL 容器空间配置器

构造函数-析构函数

构造函数，析构函数简单讲述下

构造函数初始化列表与构造函数体内复制的区别

构造函数是否可以放到 `private` 里面

构造函数，析构函数是否可以为虚函数

平凡构造函数，非平凡构造函数

移动构造函数和拷贝构造函数对比

析构函数不加 `virtual` 会发生什么，原理

深拷贝与浅拷贝区别

赋值函数，拷贝函数

C++编译过程

内存分区：全局区、堆区、栈区、常量区、代码区

空 class 有哪些函数，空 class 大小

变量声明，定义区别

指针 vs 引用

vector 的 resize，reverse 方法，迭代器什么情况下会失效（push_back 时 end 失效，扩容时 first, end 失效, 删除操作会导致删除点开始之后的迭代器全部失效）

手写 string 类

手写深拷贝，动态分配内存

虚函数作用，什么是纯虚函数，虚函数可以是静态的吗

静态函数，虚函数区别

虚函数的底层机制，虚函数指针何时生成，在多继承下怎样。虚函数表，存在哪

数组对象怎么析构，怎么释放

虚表布局，菱形继承

介绍多态，是怎么实现的

虚函数实现动态多态的原理

运行时多态，编译时多态（动态多态，静态多态（重载，重写，模板））

类的内存布局

继承的底层原理

内存溢出与内存泄漏的情况

野指针产生与避免

extern 关键字：修饰全局变量

static 关键字：修饰局部变量，全局变量，类中成员变量、类中成员函数

const 关键字：修饰变量，指针，类对象、类中成员函数

volatile 关键字：避免编译器指令优化

如何用 C 来实现抛出捕获异常

判断程序是死循环还是死锁

static_cast, dynamic_cast, const_cast, reinterpret_cast 区别

写一个拷贝构造函数？为什么引用传递而不是值传递

new/delete 与 malloc/free 区别，new 的实质，判断 new 是否成功的方法

4 种智能指针

shared_ptr 中循环引用怎么解决？（weak_ptr）

如果智能指针放到多线程中如何完成访问共享的对象

结构体中使用指针和对象有哪些好处

讲解一下动态绑定和静态绑定

B 继承 A,且有虚析构函数， A* a = new B; 中若调用虚析构函数，会调用父类 A 的还是子类 B 的析构函数

类指针如何用 C++转换类别，如 A* a 如何转换到 B*，是否所有的指针都可用 dynamic_cast 进行转换。

RAII 与 pimpl 惯用法

C++里好的编码技巧

Expression Template，奇异递归模板模式 CRTP

C++11 新特性知道哪些

std::is_same 与 std::decay

std::is_same<Type_a, Type_b>::value, 可以用于模板参数检查

严格判断类型是否一致(int, int&, const int, const int& 都不是同一种类型)

如果不需要如此严格的校验,可以利用 std::decay.

std::is_same<typename std::decay<Type_a>::type, Type_b>::value,

这个检查对于 int, int&, int&&, const int&，会判断是同一种类型;

int[2]与 int*是同种类型， int(int)与 int(*)(int)是同种类型。

auto, for_each，nullptr

std::forward, std::move, 右值，右值引用，移动构造函数

emplace_back(), push_back()

lamda 表达式

std::bind， std::function

统一的类成员初始化方法与 std::initializer_list

注解标签（attributes）

final, override,=default,=delete 语法

range-based 循环语法

结构化绑定

stl 容器新增的实用方法

std::thread

线程局部存储 thread_local

线程同步原语 std::mutex, std::condition_variable

原子操作类

智能指针类

多线程

设计线程池

设计模式

如何保证单例模式只有唯一实例，单例模式的多线程安全性。

工厂模式优点

Python

python 的 set()底层是什么数据结构

python 装饰器作用

提高 python 效率的方法

数据结构与算法

数组

链表

栈

队列

堆

二叉树：二叉搜索树，平衡树，红黑树，

红黑书与普通二叉树区别，红黑树自平衡是自动平衡吗，怎么实现自平衡

B 树、B+树

哈希表及哈希冲突

解决 hash 冲突的方法,说一下一致性 hash

快排过程/时间复杂度

二分查找

回溯法

动态规划

如何判断两个链表相交？如果两个链表有环怎么办

如何给链表排序？时间复杂度 O(nlogn),空间复杂度 O(1)

TopK 如何用堆排序实现

操作系统

堆栈的区别

有了进程，为什么还要线程

多进程与多线程比较

孤儿进程与僵尸进程

单核机器上写多线程，是否要考虑加锁

线程间共享内存，什么时候用到条件变量，什么时候用到锁

线程间同步方式：互斥锁，自旋锁，读写锁，条件变量

互斥锁与自旋锁的底层区别

死锁及避免

线程上下文切换的流程

进程上下文切换的流程

进程的调度算法

缓冲区溢出漏洞

malloc() mmap()底层实现

free()入参是 void*,如何知道大小

内存碎片是什么？有几种形式，产生的原因是什么

字节序，如何转化

(struct)内存字节对齐，为什么要字节对齐，什么因素影响字节对齐，可以 1 字节对齐吗

协程

阻塞 IO，非阻塞 IO。

同步，异步

静态链接与动态链接的过程

虚拟内存的概念

MMU 地址翻译的具体流程

缺页处理过程

缺页置换算法：最久未使用，先进先出，最佳置换

Linux

用户态，内核态，来回切换为什么效率低

Linux 进程间通信方式：PIPE,FIFO,消息队列，信号量，共享内存，socket

管道与消息队列对比

fork 进程的底层：读时共享，写时复制

gdb 如何调试堆栈

fork, wait, exec

gdb

如何启动和结束调试

如何添加，删除，启用，禁用断点（包括普通断点，条件断点，数据断点）

如何查看当前断点下的调用堆栈

如何查看程序运行过程中的线程信息

如何查看某个变量的内存值

业务

你会如何设计一个车辆管理系统

代码优化怎么做的，重构从哪些方面考虑