C++最大的问题是安全问题， 安全问题可以主要分为3个部分，指针，数组，输入输出。指针乱使用会导致两个问题，当指针管理一个资源类时被重新指向，会导致资源泄露。当多个指针指向同一个资源类，同时其中一个指针释放了资源类，而别的指针不知道的情况会，会导致指针悬空。

因便有了智能指针，智能指针有4种，auto，share，unique weak，其中auto因为自身问题被淘汰了，逐渐用unique代替。

Unique指针代表独享资源类，不支持复制操作，只支持移动操作。Unique指针可以通过移动操作转换为share指针。支持指向实现的指针。

Share指针支持复制和移动操作，share指针主要分为两部分，一部分是原始指针，一部分是控制块。控制块中通常包含3类数据，引用计数，weak计数，一些其他数据，自定义的删除器和分配器等。

使用引用计数需要记得3点，引用计数会增加内存消耗，引用计数必须动态分配，引用计数的递增递减必须是原子的。

由于控制块的存在。当多个share指针从原始指针指向同一个资源类时，会创建多个控制块，导致删除的时候，多个引用计数归零，重复删除导致未定义行为。通常使用makeshare来创建share指针。

Share不能转换为unique指针。

Weak指针可以在不参与对象生命周期的情况下观察对象。

工厂函数，观察者模式，缓存。

std：：weak\_ptr的潜在用例包括缓存、观察者列表和防止std：：shared\_ptr循环。

你可以利用智能指针来实现一种策略，比如延迟加载（lazy fetching），或者自定义删除器。