C++11的常用新特性

1、auto 自动类型推导

编译器自动推导变量的类型

2、范围for循环

for (auto& i : vec) {

}

3、Lambda表达式

匿名函数，Lambda表达式可用来捕捉外部变量，并且可以作为参数传递给函数

[捕捉列表]（参数列表）->返回类型 {

// function body  
}

4、类型别名

using Integer = int;

5、原子操作（std：：atomic）

提供无锁机制来实现线程安全的操作。

std：：atomic通过原子操作来避免数据竞争。

fetch\_add：增加并获取旧值

fetch\_sub：减少并获取旧值

load：读取原子变量的值

store：设置原子变量的值

6、nullptr

用于表示空指针，解决传统NULL常量在某些情况下可能被定义为整数0，在重载和模板场景中可能会引发问题

7、移动语义和std::move和右值引用（&&）

右值引用 &&

std：：move 将左值显式转换为右值

8、std：：thread

用于创建和管理线程

9、enum class

可以用类名来找到枚举

10、std：：chrono

是用C++11引入的时间库，来处理时间点、时间段和时钟

11、默认和删除的函数

default和delete关键字，允许显式声明默认构造函数、析构函数、拷贝构造函数和赋值运算符，也可禁止特定函数的使用

12、列表初始化

A a(1,2,3)

13、智能指针：

unique\_ptr: 只能指向一个资源，独占所有权，只能通过移动语义转移，不能管理多线程需要共享的资源

share\_ptr：允许多个指针共享同一资源，共享所有权，make\_shared创建，引用计数，每创建一个对象，引用计数加1，引用计数归0，则释放资源

std::weak\_ptr：不持有资源所有权，不影响引用计数，可以通过lock（）方法，得到shared\_ptr，

若资源已经释放则返回空指针

原始指针去创建，shared\_ptr有风险，可能会被双重释放

shared\_ptr有控制块，有强引用和弱引用，weak\_ptr会指向同一个控制块