Lesson 6

1.面向对象的第一个特征是数据抽象（data abstraction）。

2.某成员函数提供了函数的 const 重载版本，那么当使用普通非 const 对象时，调用非 const 版本，当对象为常对象，调用的是 const 版本。

3.任意对象都可以调用 const 函数，但是常对象只能调用 const 函数。（常对象是保护语义，不可以使用任何带有修改语义的函数）。

4.遇到“discards qualifiers”之类的错误，一般是由 const 调用了非 const 导致的语义冲突。

5.初始化列表是一种初始化语句，而构造函数内的赋值就是简单的赋值语句。类似于 string s = “hello” 这是初始化语句，而 s = “hello”这是赋值语句。

6.类构造函数初始化列表的初始化顺序是类内部成员声明的顺序，而不是初始化列表中排列的顺序。

7.当我们未提供任何构造函数时，系统自动合成一个默认无参数的构造函数。但是当我们提供了任意一个版本的构造函数时，系统就不会再合成这个函数。

8.使用 new 操作符产生数组的时候，调用的是无参数的构造函数。其他形式按照参数进行匹配。

9.构造函数为修改语义（semantics），所以不能为 const。

10.static 成员归整个类所共有。static 函数不与对象进行绑定，这意味着 static 函数不存在 this 指针。

11.对象的普通函数可以访问 static 成员，但是 static 函数不能访问对象的普通成员。

12.函数定义在 class 内属于 inline 函数。

13.头文件里面允许放入：

a) 类型定义

b) 宏

c) 内联函数

14.线程安全：多个线程同时运行一段代码，而能保证结果的正确性，这叫做线程安全。

15.实现线程安全的方法：

a) 互斥锁和信号量、条件变量

b) 原子操作

16.pthread_cond_wait（这个函数必须在加锁的条件下才能使用）操作的步骤：

a) 首先释放锁

b) 等待，直到接收到 signal

c) 重新抢占锁

17.生产者消费者模型中正式生产或者消费的必要条件是

a) 抢到锁

b) 等待对应的条件变量：时机成熟。

18.pthread_cond_signal 去通知一个等待该条件变量的线程，通常用来表示资源可用。而 pthread_cond_broadcast 通知等待的所有线程，通常用来表示状态的改变。

19.为何以 while 代替 if：

a) 三个消费者 wait 在 full 上。
b) 某生产者线程 produce 数据后调用了 broadcast。
c) 如果采用 if，第一个线程拿到锁，消费完数据，正常。其余两个线程发生错误。
d) 如果采用 while，后面两个线程会再次检测队列状态。
e) 另外一个错误，生产者还是调用 signal。
f) 此时一个等待线程被激活，然后尚未拿到锁，另一个线程抢到锁，直接绕过 if 语句（此时队列不为空），然后将数据消费，此时队列为空。之前的线程拿到锁，继续消费导致错误。

20.C++封装 MutexLock　不要忘记链接 pthread。

21.类的普通成员函数的函数指针，包含类名。例如 void* (Thread::)(void*)。

22.Thread 的封装问题：

a) 函数指针问题，解决方案是使用 static 方法。
b) 上述方案的缺点是线程中无法访问对象的私有数据。
c) 解决方案是把对象的 this 指针当做线程参数传入。

23.作业：封装 MutexLock、Thread、Condition，然后改写生

产者消费者。

a) 尽可能使用类封装
b) 禁止使用全局变量
c) 把同步和互斥操作放在队列里面
d) 队列只需要对外提供线程安全的函数。