Condition Variables with Predicate Exercises

Condition Variable Example

- Write a program to test the code given in the lecture
  - The main function starts a reader thread and a writer thread, in that order
- Check that the program compiles and runs as expected, even when the writer thread completes before the reader thread begins
- (The code is reproduced in the next two slides)

Condition Variable Example

```
// Global variables
condition_variable cv;                          // The condition variable instance
mutex mut;                                      // The mutex used to protect the data
string sdata {"Empty"};                         // The shared data
bool condition {false};                         // Flag

// Waiting thread
void reader() {
  unique_lock<std::mutex> lk(mut);             // Acquire lock
  cv.wait(lk, [] {return condition;});         // Lambda predicate that checks the flag
  // Notification received                     // Wake up and lock mutex
  cout << "Data is " << sdata << endl;         // Use the new value
}
```

Condition Variable Example

```
// Modyifing thread
void writer() {
    cout << "Writing data..." << endl;
    std::this_thread::sleep_for(2s);              // Pretend to be busy...
    {
        lock_guard<std::mutex> lg(mut);           // Acquire lock
        sdata = "Populated";                      // Modify the shared data
        condition = true;                         // Set the flag
    }                                             // Release the lock
    cv.notify_all();                              // Notify the condition variable
}                                                 // Release the mutex
```

Multiple Threads

- Modify the reader task to display its thread ID before and after calling wait()
- Modify the main thread so that it starts three reader threads
- Modify the writer task so that it
    - Calls notify_one() once
    - Calls notify_one() three times
    - Calls notify_all() once
- Explain the results