Write simple “ping-pong” program using wait() and notify().

Create a task that finds prime numbers, where n is the limit and provided to the constructor of the task. Create a number of these tasks and drive them using threads.

Repeat previous exercise using the different types of executors.

Modify Exercise 2 so that the task is a Callable that sums the values of all prime numbers. Create several tasks and display the results.

Create a task that sleeps for a random amount of time between 1 and 10 seconds, then displays its sleep time and exits. Create and run a quantity (given on the command line) of these tasks. Do it by using ScheduledThreadPool.

Create a class with three methods containing critical sections that all synchronize on the same object. Create multiple tasks to demonstrate that only one of these methods can run at a time. Now modify the methods so that each one synchronizes on a different object and show that all three methods can be running at once.

Write program in which two tasks use a pipe to communicate.

Modify exercise 6 from previous presentation to use explicit Lock objects.

Modify exercise 7 from previous presentation to use a BlockingQueue instead of a pipe.

Create your own ReadWriteLock (or at least simple Lock).