

code that changes the value of the boolean variable then issues a call to the `notify` method. The `notify` method wakes up only one thread. Because three threads are waiting, the thread that is awakened might not be the thread that is interested in this particular state change. Therefore, the thread that is waiting for this state change is never notified.

Use of the `notify` method is safe under two conditions:

1. When only one thread is waiting, thus guaranteeing it is awakened
2. When multiple threads are waiting on the same condition and it does not matter which thread is awakened

By contrast, the `notifyAll` method wakes up all waiting threads. Using `notifyAll` guarantees that if you have a thread that is waiting on a condition, it is awakened. In the previous example, if the code that changes the value of the boolean variable calls the `notifyAll` method, all threads waiting are awakened.

Therefore, except for the two conditions listed previously, using the `notifyAll` method is better because it guarantees that all waiting threads are awakened. Remember that awakening all threads does not mean they all acquire the lock. It simply means they all wake up and compete for the lock.

Unfortunately, the `notifyAll` method, like the `notify` method, does not provide a way to specify the order in which waiting threads are notified. The order depends on the JVM, and no guarantees are made beyond the fact that all waiting threads are awakened. Threads are not necessarily notified in their priority order. This presents a problem when you need to notify multiple threads in a particular order.

To solve this problem, you must implement the solution yourself. A general solution is available, called the “Specific Notification Pattern.”² This pattern is described in the book, *Concurrent Programming in Java*TM.³ The specific notification pattern is very useful in that it allows you to control the order in which threads are notified.

2. The “Specific Notification Pattern” was developed by Tom Cargill. A paper on this pattern with code examples of its use are at <http://www.sni.net/~cargill/jgf/9809/SpecificNotification.html>.

3. *Concurrent Programming in Java*TM, Doug Lea, Addison-Wesley, 1997.