**What is need of inter-thread communication?**

Two threads can communicate with each other, by using wait(), notify() and notifyAll() methods.

The thread which is expecting updation, is responsible to call wait() method, then immediately the thread will enter into waiting state.

The thread which is responsible to perform updation, after performing updation it is responsible to call notify() method, then waiting thread will get that notification and continue its execution, with those updated items.

**In which class wait(), notify(), notifyAll() present? Why?**

It is present in object class not in thread class.

wait(), notify(), notifyAll() method present in object class but not in thread class because thread can call these methods on any java object.

**When a thread said to have owner of an object?**

To call wait90, notify(), notifyAll() methods on any object, the thread should be the owner of that object i.e. the thread should have lock of that object.

**When the thread should can have lock of an object?**

Thread should be in synchronized area. i.e. wait(), notify(), notifyAll() should be called only in the synchronized area.

**What is IllegalMonitorStateException?**

If I am called wait(), notify(), notifyAll() method from outside synchronized method, sure I will get IllegalMonitorStateException(RE).

**What will happen when the thread calls wait() method?**

If a thread calls wait() method on any object, it immediately release the lock of that particular object and entered into **[waiting]** state.

**What will happen when the thread calls notify() method?**

If a thread calls notify() method on any object, it releases the lock of that particular object but may not immeditely. Except wait(), notify(), notifyAll() where thread releases the lock.

**Signature of wait, notify, notifyAll() method?**

**public final void wait() throws InterruptedException**

**public final void native wait() throws InterruptedException**

**public final void wait() throws InterruptedException**

**public final native void notify()**

**public final native void notify()**

**If the thread call notify() method in which state it will go?**

It will move to another waiting state to get the lock, it will come to **[runnable]** state when the waiting state thread got the lock.

**Refer ThreadCommunicationReWrittenTest.java for more details**

**Difference between notify() and notifyAll()?**

We can use notify() method to give the notification to only one waiting thread if multiple threads are waiting then only one thread will be notified and the remaining threads have to wait for further notification. Which thread will be notified we can’t expect it depends JVM.

We can use notifyAll() method to give the notification to all the waiting threads of a particular object. Even though, multiple thread notified execution will be performed one-by-one because threads require lock and only one lock is available.

**Note:**

**1. synchronized(s1) {**

**...**

**..**

**..**

**..**

**s2.wait();//invalid and we get IllegalMonitorStateException**

**}**

**2. synchronized(s1) {**

**...**

**..**

**..**

**..**

**s1.wait();//valid**

**}**