# Cooperation among Threads (OR) Inter-thread Communication

Inter-thread communication in Java is a mechanism in which a thread is paused running in its critical section and another thread is allowed to enter (or lock) in the same critical section to be executed.

It is implemented by following methods of **Object class**:

* wait()
* notify()
* notifyAll()

### **1) wait() method**

The wait() method causes current thread to release the lock and wait until either another thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has elapsed.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public final void wait()throws InterruptedException | It waits until object is notified. |
| public final void wait(long timeout)throws InterruptedException | It waits for the specified amount of time. |

### **2) notify() method**

The notify() method wakes up a single thread that is waiting on this object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation.

**Syntax:**

**public** **final** **void** notify()

### **3) notifyAll() method**

Wakes up all threads that are waiting on this object's monitor.

**Syntax:**

**public** **final** **void** notifyAll()

**Steps in Inter-thread communication:**

1. Threads enter to acquire lock.
2. Lock is acquired by a thread.
3. Now thread goes to waiting state if you call wait() method on the object. Otherwise it releases the lock and exits.
4. If you call notify() or notifyAll() method, thread moves to the notified state (runnable state).
5. Now thread is available to acquire lock.
6. After completion of the task, thread releases the lock and exits.

**Differences between wait and sleep methods:**

|  |  |
| --- | --- |
| **wait()** | **sleep()** |
| The wait() method releases the lock. | The sleep() method doesn't release the lock. |
| It is a method of Object class | It is a method of Thread class |
| It is the non-static method | It is the static method |
| It should be notified by notify() or notifyAll() methods | After the specified amount of time, sleep is completed. |

**Example:**

**class** Customer

{

**int** amount=10000;

**synchronized** **void** withdraw(**int** amount)

{

System.out.println("going to withdraw...");

**if**(**this**.amount<amount)

{

System.out.println("Less balance; waiting for deposit...");

**try**{ wait(); }**catch**(Exception e){}

}

**this**.amount-=amount;

System.out.println("withdraw completed...");

}

**synchronized** **void** deposit(**int** amount)

{

System.out.println("going to deposit...");

**this**.amount+=amount;

System.out.println("deposit completed... ");

notify();

}

}

**class** Test

{

**public** **static** **void** main(String args[])

{

**final** Customer c=**new** Customer();

**new** Thread()

{

**public** **void** run()

{

c.withdraw(15000);

}

}.start();

**new** Thread()

{

**public** **void** run()

{

c.deposit(10000);

}

}.start();

}

}

**Output:**

going to withdraw...

Less balance; waiting for deposit...

going to deposit...

deposit completed...

withdraw completed