Programming Language II
CSE-215

Prof. Dr. Mohammad Abu Yousuf

yousuf@juniv.edu

Multithreaded Programming-4

Inter-thread Communication

- If you are aware of inter-process communication then it will be easy for you to understand inter thread communication.
- Inter thread communication is important when you develop an application where two or more threads exchange some information.
- There are simply three methods (next slide) and a little trick which makes thread communication possible.

Inter-thread Communication

SN	Methods with Description
1	**public void wait()** Causes the current thread to wait until another thread invokes the notify().
2	**public void notify()** Wakes up a single thread that is waiting on this object's monitor.
3	**public void notifyAll()** Wakes up all the threads that called wait( ) on the same object.

These methods have been implemented as **final** methods in Object, so they are available in all the classes. All three methods can be called only from within a **synchronized** context.

Difference between wait and sleep?

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

```
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
```

Example: **Inter-thread Communication**

- This example (next 3 slides) shows how two thread can communicate using **wait()** and **notify()** method.

```
class Chat {
    boolean flag = false;

    public synchronized void Question(String msg) {
        if (flag) {
            try {
                wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println(msg);
        flag = true;
        notify();
    }

    public synchronized void Answer(String msg) {
        if (!flag) {
            try {
                wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        System.out.println(msg);
        flag = false;
        notify();
    }
}
```

Example: **Inter-thread Communication**

```
class T1 implements Runnable {
    Chat m;
    String[] s1 = { "Hi", "How are you ?", "I am also doing fine!" };

    public T1(Chat m1) {
        this.m = m1;
        new Thread(this, "Question").start();
    }

    public void run() {
        for (int i = 0; i < s1.length; i++) {
            m.Question(s1[i]);
        }
    }
}

class T2 implements Runnable {
    Chat m;
    String[] s2 = { "Hi", "I am good, what about you?", "Great!" };

    public T2(Chat m2) {
        this.m = m2;
        new Thread(this, "Answer").start();
    }

    public void run() {
        for (int i = 0; i < s2.length; i++) {
            m.Answer(s2[i]);
        }
    }
}
```

Example: **Inter-thread Communication**

```
public class TestThread {
    public static void main(String[] args) {
        Chat m = new Chat();
        new T1(m);
        new T2(m);
    }
}
```

Example: **Inter-thread Communication**

When above program is complied and executed, it produces following result:

```
Hi
Hi
How are you ?
I am good, what about you?
I am also doing fine!
Great!
```

Thank you