**CS 180 Problem Solving and OO Programming**

Fall 2011

Recitation Week 16: December 7-11, 2011
Concurrency revisited

We will develop two classes named `WBTest` and `WB` (WB: `W`eather `B`ulletin). The `main()` method in `WBTest` performs the following tasks:

1. Creates a WB object as follows:
   `WB` wb=new WB(filename,delay);

   where `filename` is a string denoting a file to be read, and delay is an int.

2. Starts the thread `wb`.
3. Waits for `wb` to join.
4. Gets the average temperature from `wb` and prints it. If it is positive. If negative then it prints the message "Average not available". The average is obtained by calling the getAverageTemp() method on obect `wb`.

The `WB` class has the following methods:

1. A constructor.
2. The run() method
3. The getAverageTemp() method.

**Constructor**: You need to decide what should the constructor do.
**run()** method: The run() method performs the following tasks.

1. Opens a file given by main() and reads a number that represents the current temperature. Note that data in the file is written by some other method that is not a part of this problem.
2. It then closes the file and prints the temperature in the console.
3. The above two tasks are performed continuously but with a delay of 10 seconds.
4. The run() method terminates when on three successive attempts the file is opened and is found empty or after a total of 60 seconds has elapsed since the thread was started.
5. Upon termination the run() method computes and saves in a local variable the average temperature using the temperature values read so far.

**getAverageTemp():** This method simply returns the average temperature calculated by the run() method. If it is called before the run() method has terminated then it simply returns a -1.

Do not use Thread.sleep()  for obtaining the delay in step 3 above. Instead, use the System.currentTimeMillis() to obtain the current time and write code to obtain the 10 second delay

<End of Problems for Week 16>