Linked List Exercises, CSC 143 – One Possible Solution (as a complete class)

```
/**
 * One possible solution to in-class exercise
 * Notice that there is no "List" class in these simple
 * examples.  Rather, holding a reference to the first
 * Link in the list amounts to holding a reference to
 * the entire list.  This would NOT be a good public
 * interface!  However, it allows us to work with the
 * list quite easily.
 * @version F09
 */
public class PlayWithLinkedList
{
  public static void main( String[] args ) {
    int[] x0 = {7,5,-2,8};
    Link x = makeList( x0 );
    System.out.println( listToString( x ) );

    // Warm Up: has 8?
    boolean found8 = false;
    Link p0 = x;
    while ( !found8 && p0 != null ) {
      if ( p0.num == 8 )
        found8 = true;
      p0 = p0.next;
    }
    System.out.println( "Found 8? – " + found8 );

    // 1. average (shows NaN if list is empty)
    double sum = 0;
    int count = 0;
    Link p1 = x;
    while ( p1 != null ) {
      sum += p1.num;
      count++;
      p1 = p1.next;
    }
    System.out.println( "Average: " + (sum/count) );

    // 2. link to lowest
    Link lowest = x;
    if ( x != null ) {
      Link p2 = x;
      while ( p2.next != null ) {
        p2 = p2.next;
        if ( p2.num < lowest.num )
          lowest = p2;
      }
    }
    System.out.println( listToString( lowest ) );

  }
```

```
  /**
   * 3. Make a list from the numbers in the array nums.
   * Return null if nums is null or empty.
   * @param nums the source array
   * @return a Link to the first element in the list
   */
  public static Link makeList( int[] nums ) {
    if ( nums == null || nums.length == 0 ) return null;
    Link p = new Link( nums[0], null );
    Link result = p;
    for ( int i = 1; i < nums.length; i++ ) {
      p.next = new Link( nums[i], null );
      p = p.next;
    }
    return result;
  }

  /**
   * 4. Create a String representing the list using
   * this format: {7,5,-2,8}
   * @param head the source list
   * @return a String representation
   */
  public static String listToString( Link head ) {
    if ( head == null ) return "null";
    String result = "{" + head.num;
    while ( head.next != null ) {
      head = head.next;
      result += "," + head.num;
    }
    return result + "}";
  }



  static class Link
  {
    int num;
    Link next;

    Link( int num, Link next ) {
      this.num = num;
      this.next = next;
    }
  }
}
```