```
/*
Leith Conybeare, Shantanu Jha
APCS1 pd5
HW42--Array of Titanium
2015-12-06
*/

/*****************************
 * SKELETON for
 * class SuperArray --  A wrapper class for an array.
 * Maintains functionality:
 *  access value at index
 *  overwrite value at index
 *  report number of meaningful items
 * Adds functionality to std Java array:
 *  resizability
 *  ability to print meaningfully
 *  add item (at end)
 *  insert item
 *  remove item (while maintaining "left-justification")
 *****************************/

public class SuperArray implements ListInt {

    //~~~~~INSTANCE VARS~~~~~
    //underlying container, or "core" of this data structure:
    private int[] _data;

    //position of last meaningful value
    private int _lastPos;

    //size of this instance of SuperArray
    private int _size;


    //~~~~~METHODS~~~~~
    //default constructor â€“ initializes 10-item array
    public SuperArray()
    {
      _data = new int[10];
      _lastPos = -1; //flag to indicate no lastpos yet
      _size = 0;
    }


    //output array in [a,b,c] format, eg
    // {1,2,3}.toString() -> "[1,2,3]"
    public String toString()
    {
      String foo = "[";
      for( int i = 0; i < _lastPos+1; i++ ) {
          foo += _data[i] + ",";
      }
      //shave off trailing comma
      if ( foo.length() > 1 )
          foo = foo.substring( 0, foo.length()-1 );
      foo += "]";
      return foo;
```

```
    }


    //double capacity of this SuperArray
    private void expand()
    {
      int[] temp = new int[ _data.length * 2 ];
      for( int i = 0; i < _data.length; i++ )
          temp[i] = _data[i];
      _data = temp;
      _size *= 2;
    }


    //accessor -- return value at specified index
    public int get( int index ) { return _data[index]; }


    //mutator -- set value at index to newVal,
    //           return old value at index
    public int set( int index, int newVal )
    {
      int temp = _data[index];
      _data[index] = newVal;
      return temp;
    }


    // ~~~~~~~~~~~~~~~ PHASE II ~~~~~~~~~~~~~~~
    //adds an item after the last item
    public void add( int newVal ) {
      if(_lastPos == -1){
            _data = new int [1];
            _size = 1;
            _lastPos = 0;
            _data[0] = newVal;
      }
      else {
      if(_lastPos+1==_size) expand();
      _data[_lastPos + 1] = newVal;
      _lastPos++;
      }
     }


    //inserts an item at index
    //shifts existing elements to the right
    public void addAtIndex( int index, int newVal ) {
      if(_lastPos+1==_size){
            expand();
      }
      int [] temp = new int[_size];
      int EndSize = _size - index; //how many meaningful elements are going
to be put in the temp array of size _size
      for(int i = 0; i < EndSize; i++){
            temp[i] = get(i+index);
      }
      set(index,newVal);
```

```
      for(int j = 0; j < EndSize-1; j++){
            set(j+index+1,temp[j]);
      }
      _lastPos++;
    }


    //removes the item at index
    //shifts elements left to fill in newly-empted slot
    public void remove( int index ) {
      int [] temp = new int[_size];
        int EndSize = _size - index - 1; //how many meaningful elements are
going to be put in the temp array of size _size
        for(int i = 0; i < EndSize; i++){
                temp[i] = get(i+index + 1);
        }
      for(int j = 0; j < EndSize; j++){
            set(j+index, temp[j]);
      }
      _lastPos--;
    }


    //return number of meaningful items in _data
    public int size() {
      return _lastPos + 1;
    }


    //main method for testing
    public static void main( String[] args )
    {

      ListInt mayfield = new SuperArray();
      System.out.println("Printing empty SuperArray mayfield...");
      System.out.println(mayfield);

        mayfield.add(5);
        mayfield.add(4);
        mayfield.add(3);
        mayfield.add(2);
        mayfield.add(1);

        System.out.println("Printing populated SuperArray mayfield...");
        System.out.println(mayfield);

        mayfield.remove(3);
        System.out.println("Printing SuperArray mayfield post-remove...");
        System.out.println(mayfield);
        mayfield.remove(3);
        System.out.println("Printing SuperArray mayfield post-remove...");
        System.out.println(mayfield);

        mayfield.addAtIndex(3,99);
        System.out.println("Printing SuperArray mayfield post-insert...");
        System.out.println(mayfield);
        mayfield.addAtIndex(2,88);
        System.out.println("Printing SuperArray mayfield post-insert...");
```

```
        System.out.println(mayfield);
        mayfield.addAtIndex(1,77);
        System.out.println("Printing SuperArray mayfield post-insert...");
        System.out.println(mayfield);
      //*****INSERT ANY ADDITIONAL TEST CALLS HERE*****
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
    }//end main

}//end class
```

```
/*
Leith Conybeare, Shantanu Jha
APCS1 pd5
HW42--Array of Titanium
2015-12-06
*/

public interface ListInt {

    void add( int newVal );
    //no return
    //adds newVal to beginning of array
    //right justified

    void addAtIndex( int newVal, int index );
    //adds newVal at index, shifting everything to the right
    //fails for index >= size

    void remove( int index );
    //removes value at index, shifting everything in front to the left

    int size();
    //returns the number of meaningful values in the array
    //one more than _lastPos

}
```