

# ArrayList Practice Lab Exercises\*

## Warm Up: Arrays arrayMystery

Added by: Marty Stepp

(<http://practiceit.cs.washington.edu/problem.jsp?category=University+of+Washington+CSE+143%2FCSE2+Sections%2FArrays&problem=arrays-arrayMystery>)

Consider the following method:

```
public static void mystery(int[] list) {
    for (int i = 1; i < list.length; i++) {
        list[i] = list[i] + list[i - 1];
    }
}
```

Indicate in the right-hand column what values would be stored in the array after the method `mystery` executes if the integer array in the left-hand column is passed as a parameter to `mystery`.

Be sure to enter your array in correct format, and entering elements with their appropriate type. For example, for an `int[]` holding the elements 1, 2, and 5, this would be entered as: {1, 2, 5}

{8}	<input type="text"/>
{6, 3}	<input type="text"/>
{2, 4, 6}	<input type="text"/>
{1, 2, 3, 4}	<input type="text"/>
{7, 3, 2, 0, 5}	<input type="text"/>

# ArrayList Practice Lab Exercises

## 1) 143 Practice Midterm 5 (09wi) ArrayListMystery

Added by: Eric Spishak

(<http://practiceit.cs.washington.edu/problem.jsp?category=University+of+Washington+CSE+143%2FCSE2+Exams%2FCSE2+Midterm+Exams%2F143+Practice+Midterm+5&problem=143practicemidterm5-ArrayListMystery>)

Consider the following method:

```
public static void mystery5(ArrayList<Integer> list) {
    for (int i = 0; i < list.size(); i++) {
        int element = list.get(i);
        list.remove(i);
        list.add(0, element + 1);
    }
    System.out.println(list);
}
```

Write the output produced by the method when passed each of the following ArrayLists:

[10, 20, 30]

[8, 2, 9, 7, 4]

[-1, 3, 28, 17, 9, 33]

\*All Exercises property of UW Practice It! (<http://practiceit.cs.washington.edu>), used here for Garfield HS AP CS class, March 2013.

# ArrayList Practice Lab Exercises\*

## 2) ArrayList switchPairs

Added by: Jeff Prouty

(<http://practiceit.cs.washington.edu/problem.jsp?category=University+of+Washington+CSE+143%2FCS2+Sections%2FArrayList&problem=arraylist-switchPairs> )

Write a method `switchPairs` that switches the order of values in an `ArrayList` of `Strings` in a pairwise fashion. Your method should switch the order of the first two values, then switch the order of the next two, switch the order of the next two, and so on. For example, if the list initially stores these values: {"four", "score", "and", "seven", "years", "ago"} your method should switch the first pair, "four", "score", the second pair, "and", "seven", and the third pair, "years", "ago", to yield this list: {"score", "four", "seven", "and", "ago", "years"}

If there are an odd number of values in the list, the final element is not moved. For example, if the original list had been: {"to", "be", "or", "not", "to", "be", "hamlet"} It would again switch pairs of values, but the final value, "hamlet" would not be moved, yielding this list: {"be", "to", "not", "or", "be", "to", "hamlet"}

## 3) ArrayList markLength4

Added by: Jeff Prouty

(<http://practiceit.cs.washington.edu/problem.jsp?category=University+of+Washington+CSE+143%2FCS2+Sections%2FArrayList&problem=arraylist-markLength4> )

Write a method `markLength4` that takes an `ArrayList` of `Strings` as a parameter and that places a string of four asterisks "\*\*\*\*" in front of every string of length 4. For example, suppose that a variable called `list` contains the following values: {"this", "is", "lots", "of", "fun", "for", "every", "Java", "programmer"} And you make the following call: `markLength4(list)`; then `list` should store the following values after the call: {"\*\*\*\*", "this", "is", "\*\*\*\*", "lots", "of", "fun", "for", "every", "\*\*\*\*", "Java", "programmer"}

**Notice** that you leave the original strings in the list, "this", "lots", "Java", but include the four-asterisk string in front of each to mark it.

## 4) 143 Practice Midterm 3 removeBadPairs

Added by: Marty Stepp

(<http://practiceit.cs.washington.edu/problem.jsp?category=University+of+Washington+CSE+143%2FCS2+Exams%2FCS2+Midterm+Exams%2F143+Practice+Midterm+3&problem=143practicemidterm3-removeBadPairs> )

Write a method `removeBadPairs` that accepts an `ArrayList` of integers and removes any adjacent pair of integers in the list if the left element of the pair is larger than the right element of the pair. Every pair's left element is an even-numbered index in the list, and every pair's right element is an odd index in the list. For example, suppose a variable called `list` stores the following element values:

```
[3, 7, 9, 2, 5, 5, 8, 5, 6, 3, 4, 7, 3, 1]
```

We can think of this list as a sequence of pairs: (3, 7), (9, 2), (5, 5), (8, 5), (6, 3), (4, 7), (3, 1). The pairs (9, 2), (8, 5), (6, 3), and (3, 1) are "bad" because the left element is larger than the right one, so these pairs should be removed. So the call of `removeBadPairs(list)`; would change the list to store the following element values:

```
[3, 7, 5, 5, 4, 7]
```

If the list has an odd length, the last element is not part of a pair and is also considered "bad;" it should therefore be removed by your method.

If an empty list is passed in, the list should still be empty at the end of the call. You may assume that the list passed is not `null`. You may not use any other arrays, lists, or other data structures to help you solve this problem, though you can create as many simple variables as you like.

\*All Exercises property of UW Practice It! (<http://practiceit.cs.washington.edu> ), used here for Garfield HS AP CS class, March 2013.