# CS 273 Laboratory 12A: ArrayLists

This lab gives you experience using ArrayLists and contrasting the behavior to arrays. There are 50 points possible in this lab.

### Part 1. Adding Items to Arrays and ArrayLists

### Open class ArrayLists in the Lab12A project. Note, the code is not expected to compile until you complete some of the items below.

### Implement the initList method to create an ArrayList of String and set its initial capacity to 1. As a reminder, below is the syntax to create a new ArrayList. Note, ArrayList requires a library be imported. If you do not recall which library needs to be imported, consult the Java API. ArrayList<type> listName = new ArrayList<>(initCapacity);

### In the initList method, use a for-each loop to add all the Strings in the given array into the ArrayList you created. DO NOT use the hardcoded String values from the main method. A for-each loop can be a convenient way to iterate through a collection of items like an array or an ArrayList. A for-each loop has the syntax shown below, where type is the type of values stored in the collection.

### for(type variableName : collectionName) { //do something with variableName here, like add it to //an ArrayList }

### For instance, if you wanted to loop through an array of int called intArray, you would write it like so:

### for(int someVariable : intArray) { //do something with someVariable here }

### Finally, return the ArrayList you created.

### At this point, we want to print out the contents of the ArrayList to ensure that all of the items from the array have been added correctly. The printArrayList method is already called from the main method, so you need to implement the printArrayList method. Note, you do not need a loop of any kind to print the contents of an ArrayList. Just pass the ArrayList object as a parameter to a System.out.println() call.

### Now we want to add another item to the given array, but the array is already full. Inside the addToArray method, create a new array that is one bigger than the given array and copy all the contents of the given array into the new array. Do not hardcode the lengths of either array. Add the new item as the last item in the new array.

### In the addToArrayList method, add the new item to the end of the ArrayList.

### Implement the printArray method to print the contents of the array argument. All the contents of the array should print on the same line separated by a space or comma (it’s ok if you have an extra comm at the end). Be sure to print a new line at the end of all the items.

### Uncomment the relevant portion of the main method so that when you run the program both the updated array and ArrayList print out. Was it easier to add the new item to a full array or to an ArrayList? Create a file called answers12.txt or answers12.docx and write your answer there.

### checkpoint 1 (20 points): Show your lab instructor or assistant your answer to question 7, your code, and the printed contents of your array and ArrayList.

### Part 2. Removing Items from Arrays and ArrayLists

### While it isn’t absolutely necessary, we often stack the contents of arrays so that any null elements occur only at the end of the array. This allows us to stop looping through an array when searching for something as soon as we hit a null element. Using this approach, we will remove an element from the array and adjust the contents of the rest of the array so that the array remains “stacked.”

### In the removeFromArray method, remove the given item from the array by setting the corresponding index in the given array to null. Do not hardcode the element that is removed. Your method should be general enough to handle removing any item that is passed in as an argument. Uncomment the relevant portion of the main method and run the program to print the contents of the array and verify this has been done correctly. You may even want to change the argument passed to removeFromArray to make sure any item you specify could be removed.

### Shift all the elements to the right of the null element one place to the left so that there are no null values preceding non-null values. Do not hardcode the location of the null element or assume that every element has to be shifted. Instead, you will have to check whether an element should be shifted or not. After shifting everything, the last index in the array should be set to null (otherwise the last two elements would be the same, which we don’t want). Do not use a hardcoded value to refer to the last index. Run the program to print the contents of the array and verify this has been done correctly.

### In the removeFromArrayList method, remove the given item from the ArrayList. Uncomment the relevant portion of the main method and run the program to print the contents of the ArrayList to verify this has been done correctly. Do you need to shift the elements of the ArrayList for it to remain “stacked”? (Put the answer in your “answers” file.)

### Was it easier to keep the array or ArrayList “stacked” after removing an element? (Put the answer in your “answers” file, created above.)

### checkpoint 2 (20 points): Show your lab instructor or assistant the answers to questions 10 and 11, your code, and the printed contents of your array and ArrayList.

### Part 3. Searching an ArrayList

### In order to search through an array, you need a loop. You have implemented that many times at this point in the semester, so we won’t do that again now. Here you will search through an ArrayList, but this time you won’t need iteration.

### Open the ArrayListSearch class that has been created for you.

### An ArrayList that contains the numbers 0, 10, 20, 30, 40, 50, 60, 70, and 80 in a jumbled order has been created for you. Remember, ArrayLists can only store objects, not primitives, so that’s why the ArrayList is parameterized with the Integer class, as opposed to the int type.

### Ask the user for a number an determine whether the given number exists in the ArrayList. If it does, return the index of the number. If it does not, inform the user that the number was not found. Again, you do not need a loop to search an ArrayList. Refer to the Java API if you’re unsure how ArrayLists support search.

### Is it easier to search through an array or an ArrayList? (Put your answer in your “answers” file, created above.)

### checkpoint 3 (10 points): Show your lab instructor or assistant your answer to question 4, your code, and the results of your program for various inputs, including a number that does not exist in the ArrayList.