# Lab 2

## *Goals for this Lab*

1. Write a generic array list class.

## *Task Description*

Your goal for this lab is to write a generic `ArrayList` class similar to the one in the given lecture notes on array lists.

## The ArrayList Class

The public constructors and methods required for the `ArrayList` class are listed here. The type `E` is the generic type of an element of the list.

| | |
|---|---|
| **ArrayList()** | Construct an empty `ArrayList` object. |
| **int size()** | Return the size (number of items) in this `ArrayList`. |
| **boolean isEmpty()** | Return `true` if this `ArrayList` has no items. (This is the same as the size equal to zero.) Return `false` if the size is greater than zero. |
| **void add(E value)** | Add the given element, `value`, to the end of the list. |
| **void add(int index, E value)** | Add the given element, `value`, to the list at the given `index`. After this operation is complete, `get(index)` will return `value`. This operation is only valid for `0 <= index <= size()`. |
| **E get(int index)** | Return the element of the list at the given `index`. This operation is only valid for `0 <= index < size()`. This operation does not modify the list. |
| **E remove(E value)** | Removes the first occurrence of the specified element from this list, if it is present. |
| **E remove(int index)** | Remove and return the element with the given `index` from the list. This operation is only valid for `0 <= index < size()`. After this operation, all elements that had an index greater than `index` (as determined by `get()`) will have their index reduced by one. |
| **void clear()** | Removes all the elements from this list. |

## *Requirements*

1. Your class must be named `ArrayList`.
2. Your class must provide the methods listed above for construction, accessing, and manipulating `ArrayList` objects.
3. Other than for testing purposes, your `ArrayList` class should do no input or output.

## *Testing*

This lab will be manually tested. You may want to create your own junit tests, or you may manually test your code. If manually testing you may want to create a toString() function to be able to see what your array list looks like at any one point in time.