

## Meet ArrayLists

Arrays have some pretty serious limitations. Arrays:

- are static in size, once you define their capacity they cannot grow or shrink
- store data in a rigid structure, so it is difficult to move the location of any single value
- it is difficult to insert a value into the array and maintain order
- it is difficult to remove a value from an array and reorder elements accordingly
- it is difficult to change the overall order of elements, say to sort them

## Creating an ArrayList

```
ArrayList<dataType> name = new ArrayList<dataType>();
```

```
ArrayList<String> name = new ArrayList<String>();
```

### Wrapper Classes

Primitive wrapper class

Int Integer

Boolean Boolean

Double Double

Char Character

That means to make an ArrayList of ints you would use the "Integer" wrapper class like so:

```
ArrayList<Integer> name = new ArrayList<Integer>();
```

### Storing Data with an ArrayList

```
dataType variable = name.get(index); // equivalent of name[index]
```

```
name.set(value, index); // equivalent of name[index] = value;
```

```
list.size(); // equivalent to list.length
```

What is the ArrayList equivalent to accessing an element in an array?

```
list.get(index);
```

## Useful ArrayList Methods

Methods example description

`set(index, value)`

`list.set(0, 'x')`

replaces the current value at the given index with the given value

`get(index)`

`list.get(0)`

returns the current value stored at the given index

`add(value)`

`list.add('x')`

adds the given value to the end of the list

`add(value, index)`

`list.add('x', 1)`

inserts the given value at the given index, shifting anything that was at that index or later towards the back of the list

`indexOf(value)`

`list.indexOf('x')`

returns the first index of where the given value is found, will return -1 if the value is not in the list

`contains(value)`

`list.contains('x')`

returns true if the given value is found somewhere in the list, false otherwise

`remove(index)`

`list.remove(0)`

removes and returns the value at the given index, shifting all elements after the index towards the front of the list

## ArrayLists and Loops

```
ArrayList<Integer> myList = new ArrayList<Integer>();
```

```
int staticSize = myList.size();
```

```
for (int i = 0; i < staticSize; i++) {
```

```
    if (myList.get(i) == 0) {
```

```
        myList.remove(i);
```

```
        staticSize--;
```

```
} else if (myList.get(i) == -1) {  
    myList.add(-1);  
    staticSize++;  
}  
}
```