Model 1 List Interface

In computer science, a ***list*** is a sequence of items with an ***index*** (or position) for each item.

index:	0	1	2	3	4	5	6	7	...
item:	`"Mer"`	`"Ven"`	`"Ear"`	`"Mar"`	`"Jup"`	`"Sat"`	`"Ura"`	`"Nep"`	...

Since we know which item is first, second, etc., we say that the list is ***ordered***. The first item is at the ***head*** of the list; the last item is at the ***tail*** of the list. Here is a subset of methods from `interface List<E>` in the Java API.

Modifier and Type	Method and Description
`boolean`	**`add`**`(E e)` Appends the specified element to the end of this list (optional operation).
`void`	**`add`**`(int index, E element)` Inserts the specified element at the specified position in this list (optional operation).
`E`	**`get`**`(int index)` Returns the element at the specified position in this list.
`boolean`	**`isEmpty`**`()` Returns `true` if this list contains no elements.
`E`	**`remove`**`(int index)` Removes the element at the specified position in this list (optional operation).
`E`	**`set`**`(int index, E element)` Replaces the element at the specified position in this list with the specified element (optional operation).
`int`	**`size`**`()` Returns the number of elements in this list.

Questions (15 min) Start time:

1. Give several examples of lists from everyday life.

Grocery list, playlist on Spotify, to-do lists, etc.

2. What is the index of the head of a list? In general, what is the index of the tail of a list?

The head is at index 0, the tail is at index size() - 1.

3. In Java lists, what does the `<E>` represent?

`<E>` is a generic type, i.e., the type of the list items. Notice that `add` takes an `E` as a parameter, and `get` returns an element of type `E`.

4. Of the seven methods shown from the Java API, how many change the contents of the list?

Four: `add` (at the end), `add` (at a position), `remove`, and `set`.

5. Fill in the table below to show the contents of the list after each method call. Note that the `add` method returns `true` if the list changes as a result, and the `set` method returns the element previously at the specified position.

Method Call		0	1	1	3	4		Return Value
`add(0, "A")`		A						true
`size()`		A						1
`add("N")`		A	N					true
`add(0, "R")`		R	A	N				true
`add("G")`		R	A	N	G			true
`size()`		R	A	N	G			4
`remove(0)`		A	N	G				R
`get(0)`		A	N	G				A
`get(size()-1)`		A	N	G				G
`set(1, "U")`		A	U	G				N

6. When adding or removing elements at the beginning, what did you have to do with the existing elements in the list?

To keep the list in order, existing elements need to be shifted down by one to the right for adding and to the left for removing.

7. In your own words, describe what a `List` collection is from a programmer's perspective.

A `List` is an ordered collection of elements (also known as a sequence). The user of this interface has precise control over where each element is inserted. You can access elements by their integer index and search for elements in the list.