ArrayList Objects

`ArrayList` is a generic class for representing a list of objects. It provides methods for adding, inserting, removing, getting, and setting objects. Internally, the objects are stored using an array.

Manager: Recorder:

Presenter: Reflector:

Content Learning Objectives

After completing this activity, students should be able to:

- Summarize the differences between arrays and ArrayLists.
- Describe how objects are stored in an ArrayList internally.
- Write code that manipulates an ArrayList using for loops.

Process Skill Goals

During the activity, students should make progress toward:

- Reading API documentation to learn how to use a class. (Information Processing)

Model 1 Example Code

The following examples are found in *Model1.java*. Open this file on your computer, and run the program. Record the output of each example in the space below.

```
1 int[] nums;
2 nums = new int[3];
3
4 nums[0] = 74;
5 nums[1] = 11;
6 nums[2] = 21;
7
8 System.out.println(nums.length);
9 System.out.println(nums);
```

The output is:

```
1 ArrayList<Integer> nums;
2 nums = new ArrayList<Integer>();
3
4 nums.add(74);
5 nums.add(11);
6 nums.add(21);
7
8 System.out.println(nums.size());
9 System.out.println(nums);
```

The output is:

Questions (20 min) Start time:

1. Compare the examples line by line, and summarize the differences.

a) Line 1:

b) Line 2:

c) Lines 3–6:

d) Line 8:

2. What is the main difference in the output of these two examples?

3. What happens if you add the following code after Line 6 in the array example? Verify your answer by editing *Model1.java* and running the program.

```
nums[3] = 59;
```

4. In *Model1.java*, comment out the line you just added in the previous question. Then add the following line to the ArrayList example. What is the resulting output?

```
nums.add(59);
```

5. Based on your previous answer, what ability do ArrayLists have that arrays do not?

6. Add the following line to the ArrayList example. What is the result?

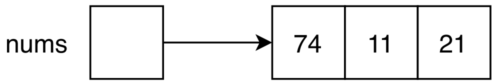
```
nums[0] = 33;
```

7. In the `ArrayList` example, is `nums` an *array* or an *object*? Justify your answer.

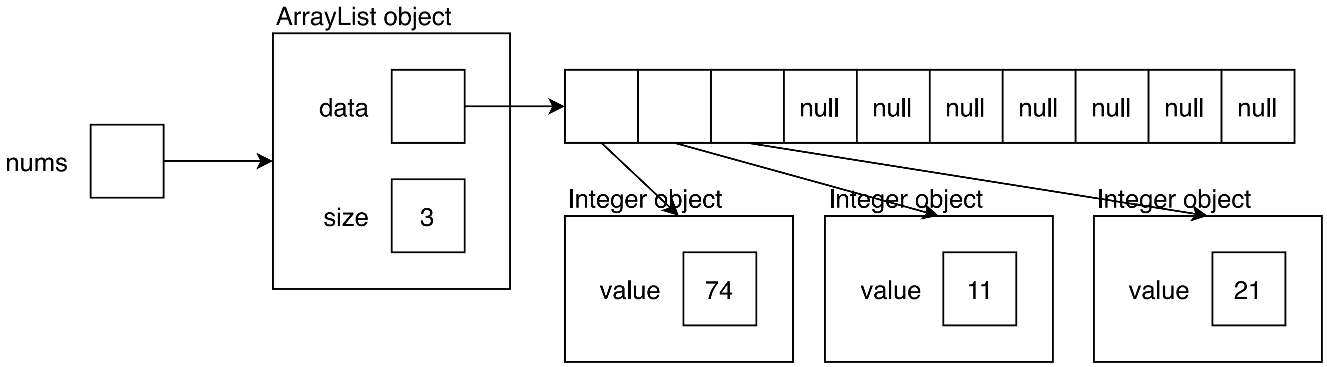
Model 2 Memory Diagrams

The following diagrams are based on the example code from Model 1.

Array of ints:



ArrayList of Integers:



Questions (10 min) Start time:

8. What is the `length` of the array inside of the ArrayList?

9. How are the contents of the `data` array different from the array of ints?

10. What happens when a fourth element is added to the ArrayList?

11. Explain how an ArrayList can “grow” when adding new elements.

12. Why do ArrayLists require so much more memory than arrays?

Model 3 ArrayList Methods

Open the ArrayList documentation to answer questions about the following methods. The data type "E" refers to the type of elements in the ArrayList (e.g., Integer).

Return Type	Method	Description
boolean	add(E e)	Appends the specified element to the end of this list.
void	add(int index, E element)	Inserts the specified element at the specified position in this list.
E	get(int index)	Returns the element at the specified position in this list.
E	set(int index, E element)	Replaces the element at the specified position in this list with the specified element.
int	size()	Returns the number of elements in this list.

Questions (15 min) Start time:

13. What value does the add method return?

14. What value does the set method return?

15. What happens to existing elements when adding an element at a specified index?

16. What are the contents of nums after running the following code?

```
ArrayList<Integer> nums;
nums = new ArrayList<Integer>();
nums.add(74);
nums.add(21);
nums.add(0, 11);
nums.set(1, 59);
```

17. The following program, found in *Model3.java*, uses an array of ints. Rewrite the program to use an ArrayList instead.

```
import java.util.Arrays;

public class Model3 {

    public static void main(String[] args) {
        final int N = 4;
        int[] nums = new int[N];
        for (int i = 0; i < N; i++) {
            nums[i] = i + 1;
        }
        for (int i = 0; i < N; i++) {
            nums[i] *= 5;
        }
        System.out.println(Arrays.toString(nums));
    }

}
```