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: **Alex Wang**

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[O] = 74;
5 nums[1] = 11;
6 nums[2] = 21;
7
8  System.out.println(nums.length);
9 System.out.println(nums);
```

The output is:

3
[I@{some memory location}

```
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:

3
[74, 11, 21]

Questions (20 min) Start time:

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

a) Line 1:

Right example uses ArrayList instead of a basic array of ints

b) Line 2:

Right example uses ArrayList instead of a basic array of ints

c) Lines 3–6:

Right example uses .add() method instead of assigning a value to an array index

d) Line 8:

Right example uses .size() method instead of .length variable

2. What is the main difference in the output of these two examples?
The left example will print basically garbage for line 9, while the right example will print the values

inside the ArrayList.

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;
```

Index 3 out of bounds for length 3

3. 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);
```

4
[74, 11, 21, 59]

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

They can increase their length on demand

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

```
nums[O] = 33;
```

"O" is not a valid array index

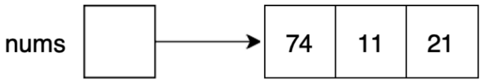
6. In the ArrayList example, is nums an *array* or an *object*? Justify your answer.

An object; arrays have the property that their length is immutable and ArrayList contradicts that property.

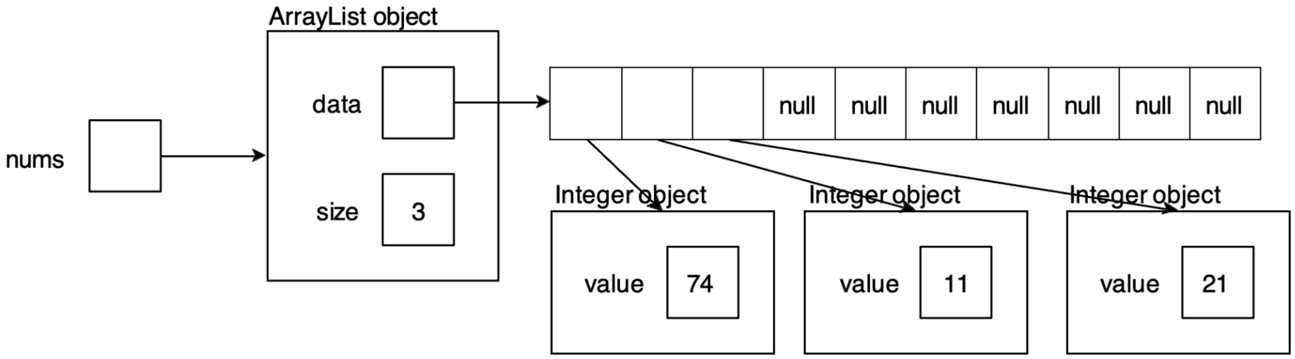
Model 2 Memory Diagrams

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

Array of ints:



ArrayList of Integers:



Questions (10 min) Start time:

7. What is the length of the array inside of the ArrayList?

10

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

The data array has references to Integer objects, while the array of ints just contains int primitives.

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

The fourth item in the data array inside the ArrayList is populated.

10. Explain how an ArrayList can "grow" when adding new elements.

New ArrayLists are copied and recreated to prevent the data array inside the ArrayList from hitting its maximum length.

11. Why do ArrayLists require so much more memory than arrays?
They store way more stuff!

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:

12. What value does the add method return?

A boolean expressing whether the passed item was added or not.

13. What value does the set method return?

The new value of the element that was updated.

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

The elements at and to the "right" of the new element are all shifted one position to the right (i.e., their indices are incremented by 1).

15. 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);
```

11, 59, 21

16. 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 = O; i < N; i++) { nums[i] = i + 1;
        }
        for (int i = O; i < N; i++) { nums[i] *= 5;
        }
        System.out.println(Arrays.toString(nums));
    }

}
```

```
Import java.util.ArrayList;
public class Model3 {
public static void main(String[] args) {
ArrayList<Integer> nums = new ArrayList<Integer>();
for (int i = 0; i <nums.size(); i++) {
nums.set(i, i+1);
}

for (int i = 0; i < nums.size(); i++) {
nums.set(i, nums.get(i) * 5);
}

System.out.println(nums);
}
}
```