Unit 6 - Lesson 4
ArrayLists

Computer Science A

Warm Up

The Joyful Pastries Food Truck owner wants cashiers to enter customer orders into a list as they are received so the baker can complete the orders from the list.

✅ **Do This:** Write pseudocode to help the owner with this problem.

💡 Discuss:

What if we don't know how many orders the owner needs to enter into the program?

What would we need to change in our solution?

Activity

🎯 Lesson Objectives

By the end of this lesson, you will be able to . . .

- Identify similarities and differences between using a one-dimensional (1D) array and an **`ArrayList`**
- Use methods in the **`ArrayList`** class to obtain the size and add elements

🚀 Question of the Day

Why would I use an **`ArrayList`** instead of an array?

Predict and Run

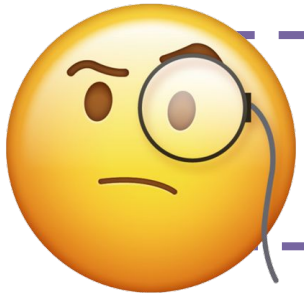
Navigate to Lesson 4, Level 1

✅ Do This:

1. Predict the output of the program

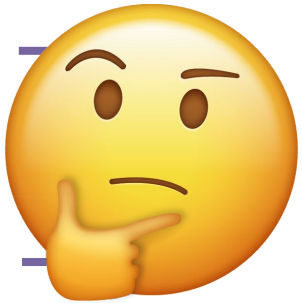
   There are no wrong answers!

2. Run it to compare your prediction with the results



What did you notice about the code in this program?

What do you wonder about the code in this program?

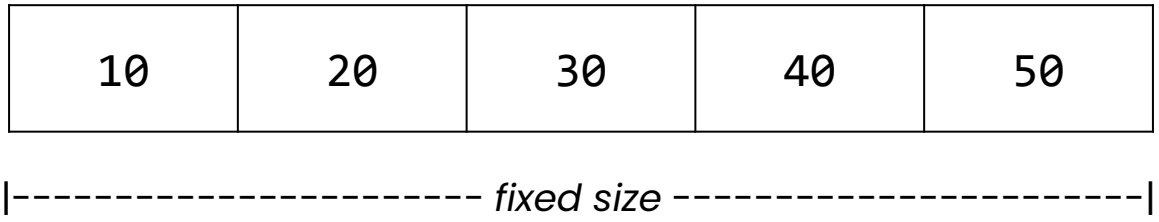


🎥 The ArrayList Class

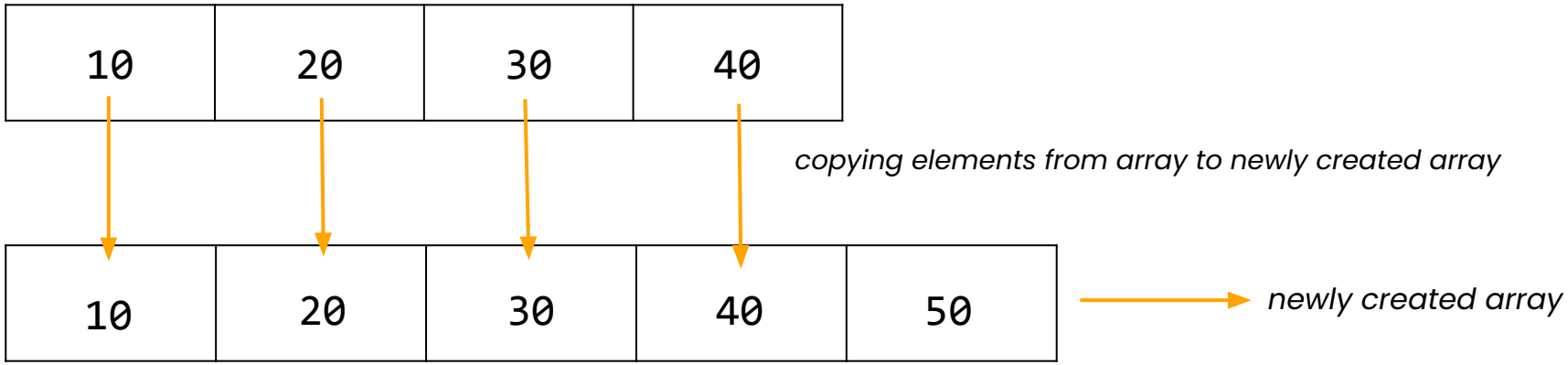
How is an **`ArrayList`** different from a 1D array?

Complete the guided notes on the 📝 **Unit 6 Guide**.

A **static data structure** is a data structure that is **fixed in size**. For example, a 1D or 2D array.



A **dynamic data structure** is a data structure that **grows** and **shrinks** as needed. For example, an **`ArrayList`**.



Investigate and Modify

Navigate to Lesson 4, Level 2

✅ Do This:

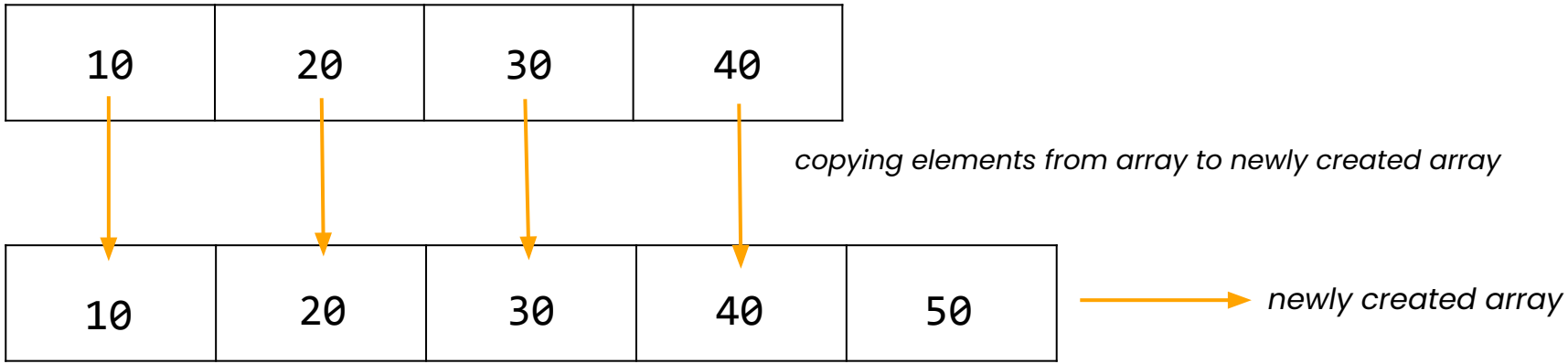
1. Investigate the code on **Levels 2 through 4**
2. Make changes as prompted and observe the results

What did you discover from the modifications you made to the code?

💡 Discuss:

When would we want to use an **`ArrayList`** instead of a 1D array?

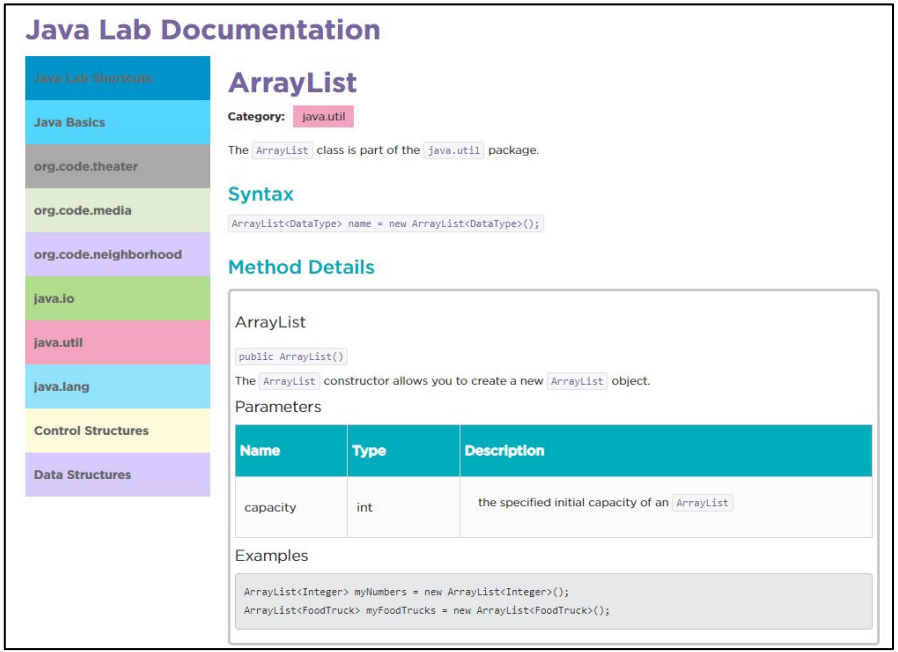
An **`ArrayList`** is a class that represents a **resizable list**.



`ArrayLists` are **mutable** . This means that the number of items they store **can change** after the list has been initialized.

The **`ArrayList`** class is part of the **`java.util`** package.

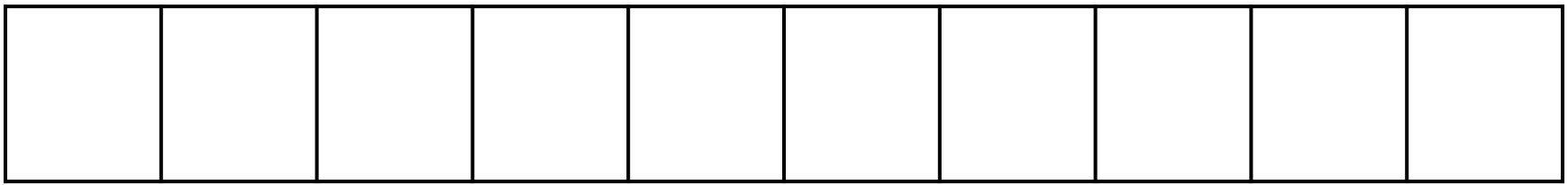
To use an **`ArrayList`** in our programs, we need to import **`java.util.ArrayList`**.



To create an **ArrayList**, we call its constructor.

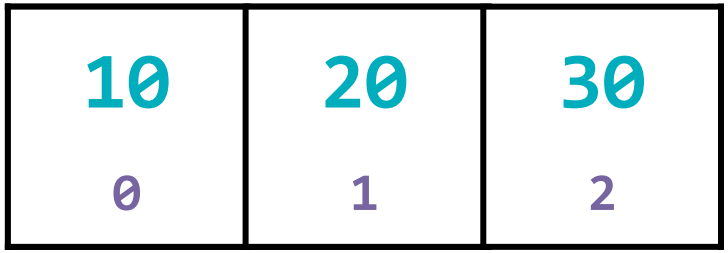
```
ArrayList<Integer> numbers = new ArrayList<Integer>();
```

This constructor creates a new empty **ArrayList** that can store **Integer** objects.

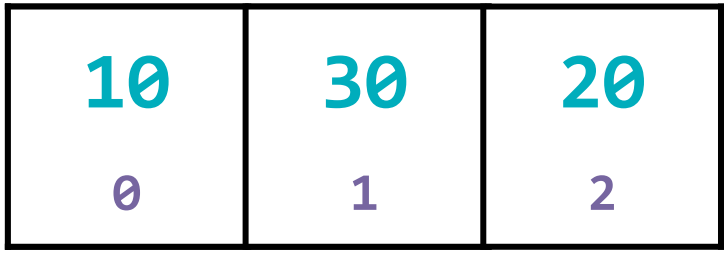


The **`ArrayList`** class has two versions of the **`add()`** method.

`numbers.add(30);` adds the element to the end of the **`ArrayList`**



`numbers.add(1, 30);` adds the element at index **`1`** of the **`ArrayList`** and shifts the rest of the elements to the right by one position



The **`ArrayList`** class also has a **`size()`** method that returns the **number of elements** in the list.

ArrayList Source Code

```
84 public class ArrayList<E> extends AbstractList<E>
85  implements List<E>, RandomAccess, Cloneable, Serializable
86 {
        ...
92   /**
93    * The default capacity for new ArrayLists.
94    */
95   private static final int DEFAULT_CAPACITY = 10;
96
97   /**
98    * The number of elements in this list
99    * @serial the list size
100   */
101   private int size;
102
103  /**
104   * Where the data is stored.
105   */
106  private transient E[] data;
```

Instance Variables

What is the purpose of each instance variable?

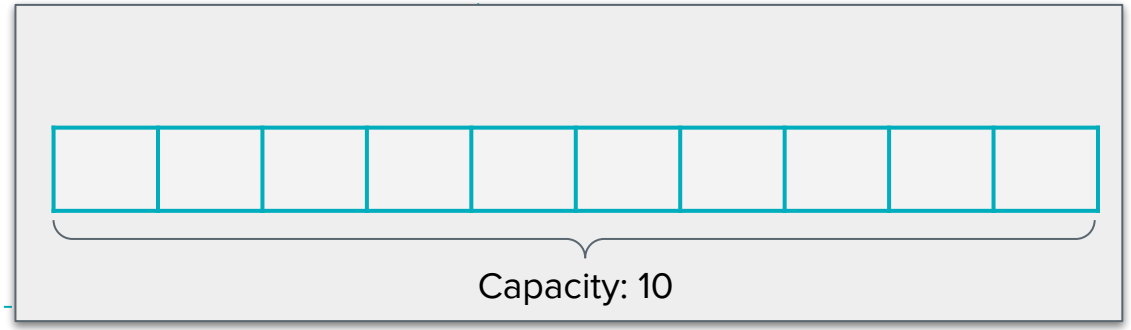
What is the relationship between an array and an **`ArrayList`**?

ArrayList Source Code

```
84 public class ArrayList<E> extends AbstractList<E>
85  implements List<E>, RandomAccess, Cloneable, Serializable
86 {
        ...
92   /**
93    * The default capacity for new ArrayLists.
94    */
95   private static final int DEFAULT_CAPACITY = 10;
96
97   /**
98    * The number of elements in this list
99    * @serial the list size
100   */
101   private int size;
102
103  /**
104   * Where the data is stored.
105   */
106  private transient E[] data;
```

Instance Variables

DEFAULT_CAPACITY is the initial length of a new array to store the elements of the **ArrayList**.

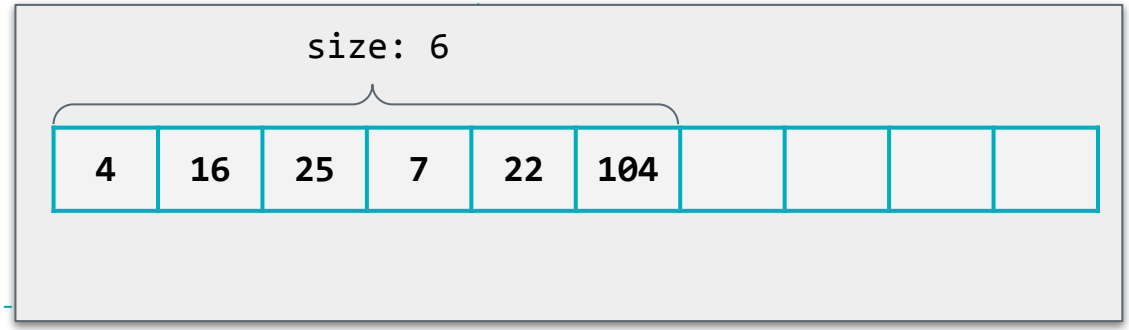


ArrayList Source Code

```
84 public class ArrayList<E> extends AbstractList<E>
85  implements List<E>, RandomAccess, Cloneable, Serializable
86 {
        ...
92   /**
93    * The default capacity for new ArrayLists.
94    */
95   private static final int DEFAULT_CAPACITY = 10;
96
97   /**
98    * The number of elements in this list
99    * @serial the list size
100   */
101   private int size;
102
103  /**
104   * Where the data is stored.
105   */
106  private transient E[] data;
```

Instance Variables

The **size** of an **ArrayList** is determined by the **number of elements** it contains.



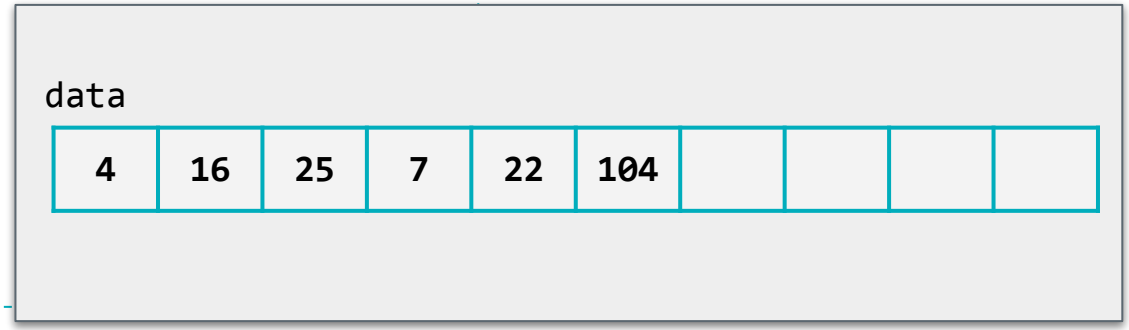
ArrayList Source Code

```
84 public class ArrayList<E> extends AbstractList<E>
85  implements List<E>, RandomAccess, Cloneable, Serializable
86 {
        ...
92   /**
93    * The default capacity for new ArrayLists.
94    */
95   private static final int DEFAULT_CAPACITY = 10;
96
97   /**
98    * The number of elements in this list
99    * @serial the list size
100   */
101   private int size;
102
103  /**
104   * Where the data is stored.
105   */
106  private transient E[] data;
```

Instance Variables

The array **data** stores the elements of the **ArrayList** in memory.

The type of **data** is **E** because the type of the array depends on the type of the **ArrayList**.



🎉 Practice

Navigate to Lesson 4, Level 5

✅ Do This:

1. **Level 5 -** Check for Understanding
2. **Level 6 -** Practice declaring and initializing an **`ArrayList`**
3. **Level 7 -** Practice adding elements to an **`ArrayList`**

Wrap Up

💡 Discuss:

How could we use an **`ArrayList`** to solve this problem?

The Project Mercury Pastries Food Truck owner wants cashiers to enter customer orders into a list as they are received so the baker can complete the orders from the list.

🎉 Today, you learned about . . .

- Similarities and differences between using a one-dimensional (1D) array and an **`ArrayList`**
- Using methods in the **`ArrayList`** class to obtain the size and add elements

🚀 Question of the Day

Why would I use an **`ArrayList`** instead of an array?

💼 Key Vocabulary

- **static data structure:** a data structure that is fixed in size
- **dynamic data structure:** a data structure that grows and shrinks as needed
- **mutable:** the ability to change after initialization