Unit 6 - Lesson 8
Removing Elements

Computer Science A

Warm Up

The Social Media Dilemma

You are writing an algorithm for your favorite social media app. This algorithm uses an array to store the names of every user someone is following. We know that people sometimes "break up" with friends for whatever reason.

We need a way to remove (or unfollow) a user from our list.

💡 Discuss:

What **challenges** did you run into with using an **array** for this problem?

What would you **like to be able to do** that an array **doesn't allow**?

Retrieve
your knowledge and ideas and write it down silently

Pair
up with a neighbor and talk about your reflections

Share
your thoughts in a class discussion

💡 Discuss:

Given what you know about the **`add()`** method for an **`ArrayList`**, how do you think an **`ArrayList`** handles removing items?

Activity

🎯 Lesson Objectives

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

- Explain the cause of a **`ConcurrentModificationException`**
- Use methods in the **`ArrayList`** class to remove elements

🚀 Question of the Day

How is removing data from an **`ArrayList`** different from removing data from an array?

Predict and Run

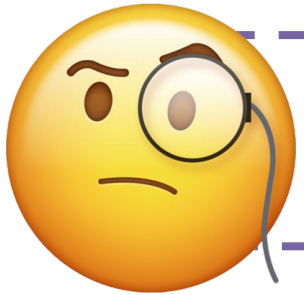
Navigate to Lesson 8, Level 1

✅ Do This:

1. Predict the output of the program

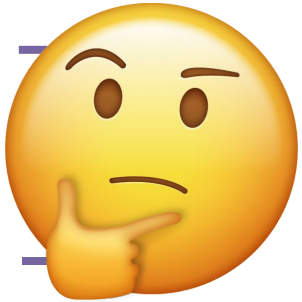
   There are no wrong answers!

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?



Investigate and Modify

Navigate to Lesson 8, 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?

The **E `remove()`** method removes the element at position **`index`**, moving the elements at position **`index + 1`** and higher to the left and subtracts **`1`** from size. The element that was at position **`index`** is returned.

```
ArrayList<String> teamList = new ArrayList<String>();
teamList.add("Falcons");
teamList.add("Bears");
teamList.add("Titans");
System.out.println(teamList);
String result = teamList.remove(1);
System.out.println(result);
System.out.println(teamList);
```

```
[Falcons, Bears, Titans]
Bears
[Falcons, Titans]
```

💡 Discuss:

How is the **`remove()`** method different from how we would remove items from an array?

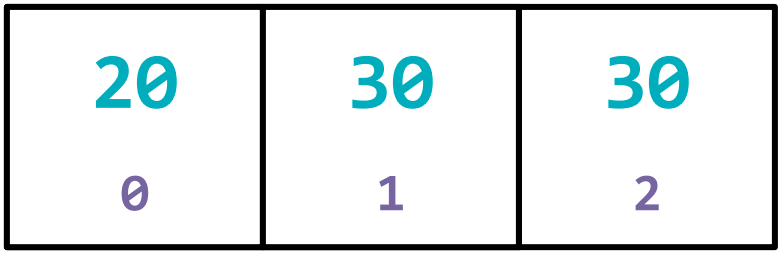
🎥 Removing Data from an ArrayList

How might the **`remove()`** method be useful in our Social Media Dilemma?

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

```
for (int index = 0; index < myList.size(); index++) {
  myList.remove(index);
}
```

index	1



```
for (int index = 0; index < myList.size(); index++) {
  myList.remove(index);
  index--;
}
```

index	0

30	30	30
0	1	2

💡 Discuss:

What differences do you notice between these two pieces of code?

Enhanced `for` Loop: Arrays

```
for (int price : prices) {
    System.out.println(price);
}
```

Enhanced `for` Loop: `ArrayList`

```
for (Integer price : prices) {
    System.out.println(price);
}
```

Changing the size of an **`ArrayList`** while traversing it using an enhanced **`for`** loop can result in a **`ConcurrentModificationException`** being thrown.

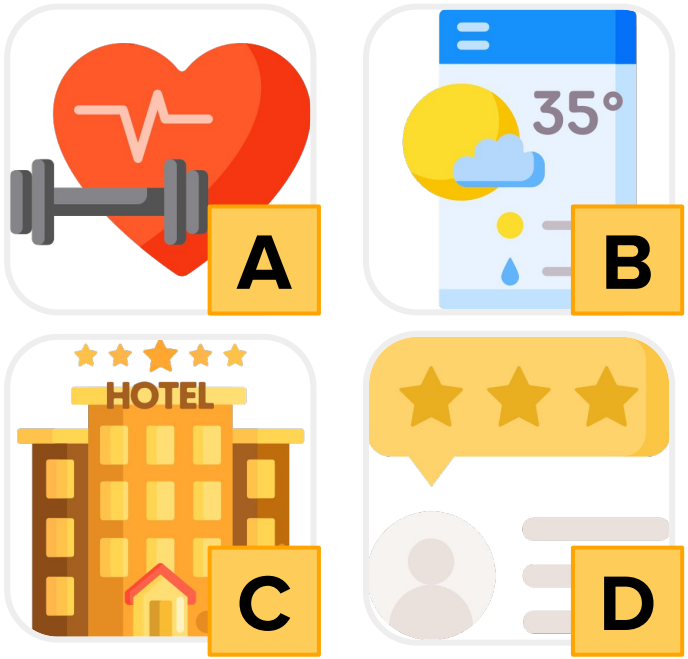
```
ArrayList<Integer> numbers = new
      ArrayList<Integer>();

numbers.add(10);

numbers.add(20);

numbers.add(30);

for (int num : numbers) {

  System.out.println(numbers);

  numbers.add(50);

}
```

```
10
ConcurrentModificationException
```

⚠️ When using an enhanced **`for`** loop with an **`ArrayList`**, you should **not** add or remove elements.

📝 **Unit 6 Guide**

🎉 Practice



Navigate to Lesson 9, Level 5

✅ Do This:

1. **Level 5 –** Check for Understanding
2. **Level 6 –** Debug a program that results in a **`ConcurrentModificationException`**
3. **Level 7 –** Practice using the **`remove()`** method to remove elements from an **`ArrayList`**

Wrap Up

💡 Discuss:

How would we solve our Social Media Dilemma using an **`ArrayList`** instead of an array? Is this problem easier to solve with an array or an **`ArrayList`**? Why?

🎉 Today, you learned about . . .

- Explain the cause of a **`ConcurrentModificationException`**
- Use methods in the **`ArrayList`** class to remove elements

🚀 Question of the Day

How is removing data from an **`ArrayList`** different from removing data from an array?