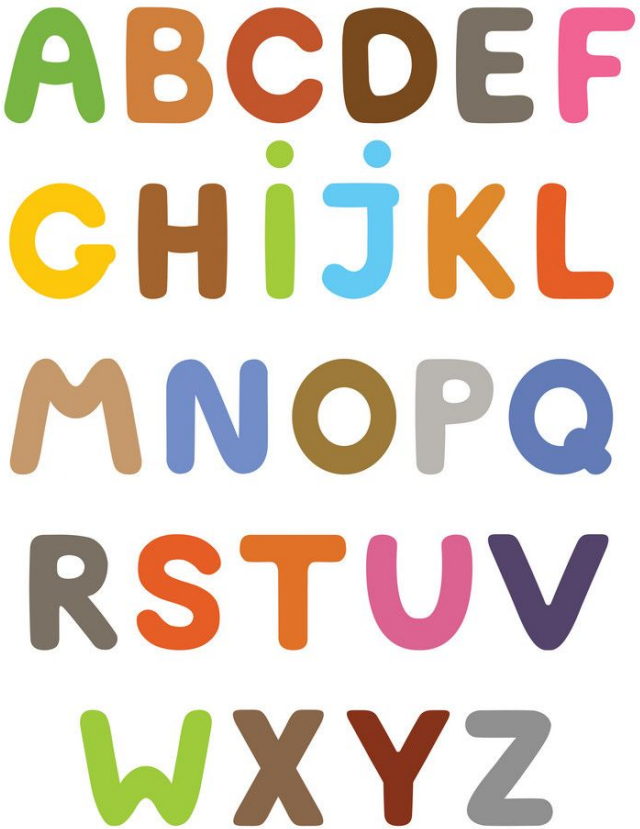
Unit 6 - Lesson 6
Comparing Strings

Computer Science A

🚀 Question of the Day

How can I determine if a list of **`String`** objects are in alphabetical order?



Lexicographical order means placing words in alphabetical order.

📝 **Unit 6 Guide**

📝 **Unit 6 Guide**

`int compareTo(String anotherString)` returns a **negative** integer if ***this*** **`String`** comes *before* the argument **`String`**, a **positive** integer if ***this*** **`String`** comes *after* the argument **`String`**, and **`0`** if the **`String`** objects contain the *same characters*.

```
String firstWord = "Hello";
String secondWord = "HELLO";
String thirdWord = "Java";
System.out.println(firstWord.compareTo(thirdWord));
System.out.println(firstWord.compareTo(secondWord));
System.out.println(thirdWord.compareTo(firstWord));
```

```
-2
32
2
```

The **`compareTo()`** method is case-sensitive, so **`"Hello"`** and **`"HELLO"`** are not considered equal.

values that are returned are based on the ode value of each character in the **`String`**.

Self Check

What will be printed after this code segment is executed?

```
String str1 = "apple";
String str2 = "banana";
System.out.println(str1.compareTo(str2));
```

A 0

B 1

C -1

D "apple"

E "banana"

💼 Key Vocabulary

- **lexicographical order:** placing words in alphabetical order

Unit 6 - Lesson 7
Lists of Objects

Computer Science A

🚀 Question of the Day

Why would I use generic types?

A **generic type** allows a class, or type, to be used as the parameter to an **`ArrayList`** and is indicated by **`< >`**.

`ArrayList`

This is the non-generic version.

In the non-generic version, you can add any type of **`Object`**.

It can lead to errors, and you have to use casting to have each element treated as its specific type.

`ArrayList<E>`

This is the generic version with a type parameter **`E`**.

You can only add a specific type of **`Object`** based on the type it is initialized with.

It minimizes errors, and you don't have to use casting.

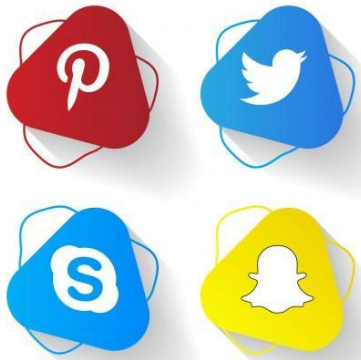
💼 Key Vocabulary

- **generic type:** allows a class, or type, to be used as the parameter to an `ArrayList` and is indicated by < >

Unit 6 - Lesson 8
Removing Elements

Computer Science A

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.

🚀 Question of the Day

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

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.

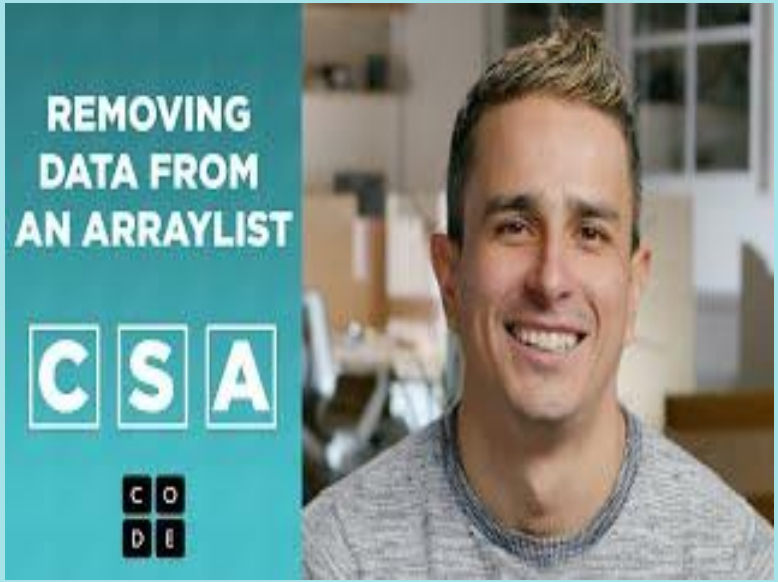
```
ArrayList<String> teamList = new ArrayList<String>();
teamList.add("Falcons");
teamList.add("Bears");
teamList.add("Titans");
System.out.println(teamList);
String result = teamList.remove(1);
System.out.println(result);
System.out.println(teamList);
```

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

🎥 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

20	30	30
0	1	2

```
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(num);
  numbers.add(50);
}
```

```
10
ConcurrentModificationException
```

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

Unit 6 - Lesson 9
ArrayList and String Algorithms

Computer Science A

🚀 Question of the Day

How can I use what I know about object-oriented programming and **`ArrayLists`** to plan and implement algorithms?

Text segmentation is the process of dividing text into words, sentences, or topics.

My	favorite	food	is	pizza

.

Text segmentation is used in NLP to . . .

- Identify and categorize named entities, such as people, organizations, and locations
- Identify the most relevant information in response to a search query
- Identify the most important sentences in a document to summarize its key points

💡 Discuss:

- What separates the **end** of one **word** from the **beginning** of the **next**?
- What separates the **end** of one **sentence** from the **beginning** of the **next**?

My favorite food is pizza.

Keyword extraction is a natural language processing technique used to find the **most used** or **most important** words in text.

I'm in love with the app! It's amazing!! The mobile version works just as well as the web version. You can create pages and control how your content is displayed very easily as the app has very intuitive and simple controls

TAG	VALUE
KEYWORD	love
KEYWORD	app
KEYWORD	mobile version
KEYWORD	web version
KEYWORD	page
KEYWORD	content
KEYWORD	simple control

Part-of-speech tagging is another natural language processing technique that **categorizes words** based on its **part of speech**.

Why	not	tell	someone	?
adverb	adverb	verb	noun	punctuation mark, sentence closer

This technique is often used to identify **names** and **places** or for **speech recognition**.

A **stop word** is a word that is filtered out of a list before or after processing text.

to
from
of
with
the
at
in
for
on

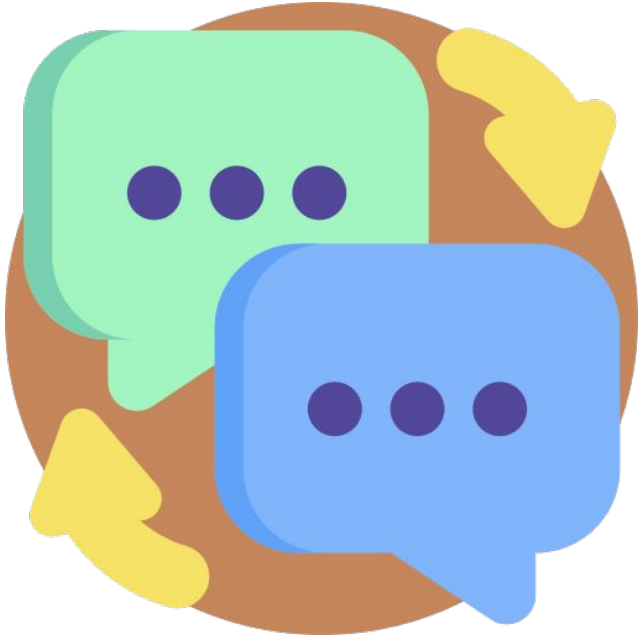
They provide little meaning to the overall text.

Stop word removal is the process of removing commonly used words from a list before or after processing text.

NLP techniques ~~can~~ help find meaning ~~in the~~ text ~~for the~~ computer ~~to~~ understand.

📝 Unit 6 Guide

Natural Language Processing Project



Use your knowledge of object-oriented programming, **`ArrayList`**s, the **`String`** class, and algorithms to create a program that uses natural language processing techniques:

- Create **`ArrayList`**s
- Implement one or more algorithms
- Use methods in the **`String`** class
- Use at least one NLP technique
- Document your code

💼 Key Vocabulary

- **text segmentation:** the process of dividing text into words, sentences, or topics
- **stop word:** a word that is filtered out of a list before or after processing text
- **stop word removal:** the process of removing commonly used words from a list before or after processing text