LEC 03

CSE 122

ArrayList

Questions during Class?

Raise hand or send here

sli.do #cse122

BEFORE WE START

Talk to your neighbors:

Coffee or tea? Or something else?

Music: 122 26sp Lecture Jams

Instructor: Elba Garza

TAs:

David	Caleb	Cole	Yang
William	Neha	Blake R.	Cady
Dani	Wesley	Carson	Diya
Rohan	Isis	Sushma	
Andrew	Colin	Connor	
Ava	Naomi	Mahima	
Shreyank	Hanna	Nicolae	
Nicole	Blake P.	Ivory	

Lecture Outline (1/5)

- Announcements
- Movie Ratings
- Using `PrintStream` for File Output
- ArrayList Recap
- ArrayList Examples

Announcements

- Programming Assignment 0 (P0) out later today!
  - Due next Thursday, April 16th!
  - Focused on File I/O
  - TODO Reflection changes!
- Creative Project 0 (C0) was due last night. How'd it go?
  - Expect grades back about a week after the assignment was due
  - Joined class late? **Use Resubmission Cycle 0 to submit it!**
- Quiz 0 is next Thursday (Apr 16th)!
  - There will be an Ed post later tonight with instructions and logistics
  - Practice Quiz to be released this weekend (and solutions posted on Monday)

Lecture Outline (2/5)

- Announcements
- Movie Raitings
- Using `PrintStream` for File Output
- ArrayList Recap
- ArrayList Examples

Movie Ratings (1/2)

In this program, we'll be examining and printing data from a file of IMDb ratings for popular U.S. movies. This will happen through 3 major user-entered commands:

(F)ind movie, **(A)dd** a rating, and **(P)rint** ratings.

small.tsv

```
5
Title   Average Total
Pride_&_Prejudice   7.8 323647
Barbie  6.9 455488
Oppenheimer 8.4       588723
Poor_Things 8.5       20542
Spider-Man:_Across_the_Spider-Verse 8.6     329200
```

Movie Ratings (2/2)

```
Welcome to the CSE 122 Movie Rating Program!
Loaded 5 movies from small.tsv!

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: F
What's the name of the movie? Pride_&_Prejudice
Movie Pride_&_Prejudice found!
    Average Rating: 7.8
    Total Ratings: 323647

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: A
What movie would you like to add your rating to? Pride_&_Prejudice
And what rating would you like to give? 100000

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: f
What's the name of the movie? Pride_&_Prejudice
Movie Pride_&_Prejudice found!
    Average Rating: 8.1
    Total Ratings: 323648

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: P
5
Title Average Total
Pride_&_Prejudice 7.8 323647
Barbie 6.9 455488
Oppenheimer 8.4 588723
Poor_Things 8.5 20542
Spider-Man:_Across_the_Spider-Verse 8.6 329200

Menu: (F)ind movie, (A)dd rating, (P)rint ratings, (Q)uit
Enter your choice: q
Thank you for using this program! Bye!
```

small.tsv

```
5
Title   Average Total
Pride_&_Prejudice  7.8 323647
Barbie  6.9 455488
Oppenheimer 8.4  588723
Poor_Things 8.5  20542
Spider-Man:_Across_the_Spider-Verse 8.6  329200
```

```
[Pride_&_..., Barbie, Oppenheimer, Poor_Things, Spider-Man…]
[7.8,         6.9,    8.4,         8.5,         8.6]
[323647,      455488, 588723,      20542,       329200]
```

Movie Ratings: Development Strategy

1. Fill in the main method with a loop that calls a method to read the data in from the .tsv file and allows the user to select between the different options (find, add, print, quit). I was nice and did this for you 😇
2. Implement a method to **load** the movie rating data from the file and populate the appropriate arrays.
3. Implement a method that allows users to **find** the rating for a movie.
4. Implement a method that allows users to **add** a rating for a movie.
5. Implement a method that allows users to **print** the movie rating data. I was nice and did this for you 😇
6. Modify the above **print** to output to a file instead!

Lecture Outline (3/5)

- Announcements
- Movie Ratings
- Using `PrintStream` for File Output
- ArrayList Recap
- ArrayList Examples

(PCM) PrintStreams for output

`PrintStream` is defined in the `java.io` package

```
import java.io.*;
```

```
File outputFile = new File("out.txt");
PrintStream output = new PrintStream(outputFile);
```

PrintStream Methods	Description
`print(…)`	Prints the given value to the set output location.
`println(…)`	Prints the given value to the set output location, and then terminates the line.

```
System.out.print("Hello, world! ");
System.out.println("#1 Bee Movie fan!");
```

```
output.print("Hello, world! ");
output.println("#1 Bee Movie fan!");
```

Hello, world! #1 Bee Movie fan!

Movie Ratings

In this program, we'll be examining and altering data from a file of IMDb ratings for popular U.S. movies. This will happen through 3 major user-entered commands:

(F)ind movie, **(A)dd** a rating, and **(S)ave** to a file.

Instead of printing to console, we will now output to a file!

Movie Ratings

```
Welcome to the CSE 122 Movie Rating Program!
Loaded 5 movies from small.tsv!

Menu: (F)ind movie, (A)dd rating, (S)ave, (Q)uit
Enter your choice: F
What's the name of the movie? Pride & Prejudice
Movie Pride_&_Prejudice found!
    Average Rating: 7.8
    Total Ratings: 323647

Menu: (F)ind movie, (A)dd rating, (S)ave, (Q)uit
Enter your choice: A
What movie would you like to add your rating to? Pride & Prejudice
And what rating would you like to give? 100000

Menu: (F)ind movie, (A)dd rating, (S)ave, (Q)uit
Enter your choice: f
What's the name of the movie? Pride & Prejudice
Movie Pride_&_Prejudice found!
    Average Rating: 8.1
    Total Ratings: 323648

Menu: (F)ind movie, (A)dd rating, (S)ave, (Q)uit
Enter your choice: S
What's the name of the file you'd like to save to? out.txt

Menu: (F)ind movie, (A)dd rating, (S)ave, (Q)uit
Enter your choice: q
Thank you for using this program! Bye!
```

small.tsv

```
5
Title   Average Total
Pride_&_Prejudice  7.8 323647
Barbie  6.9 455488
Oppenheimer 8.4  588723
Poor_Things 8.5  20542
Spider-Man:_Across_the_Spider-Verse 8.6  329200
```

```
[Pride_&_..., Barbie, Oppenheimer, Poor_Things, Spider-Man…]
[7.8,         6.9,    8.4,         8.5,         8.6]
[323647,      455488, 588723,      20542,       329200]
```

Lecture Outline (4/5)

- Announcements
- Movie Ratings
- Using `PrintStream` for File Output
- **ArrayList Recap**
- ArrayList Examples

ArrayList (1/4)

ArrayLists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - ArrayLists can only hold Objects, so might need to use "wrapper" types: `Integer, Double, Boolean, Character, etc.`

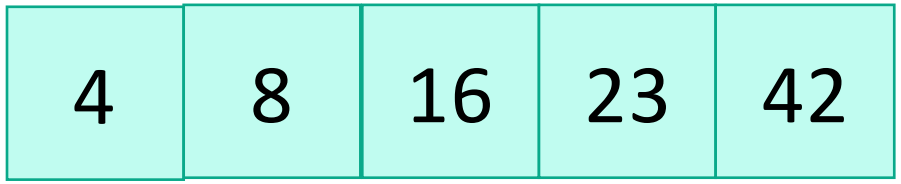
But ArrayLists have dynamic length (so they can resize!)

ArrayList (2/4)

ArrayLists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - ArrayLists can only hold Objects, so might need to use "wrapper" types: `Integer, Double, Boolean, Character, etc.`

But ArrayLists have dynamic length (so they can resize!)



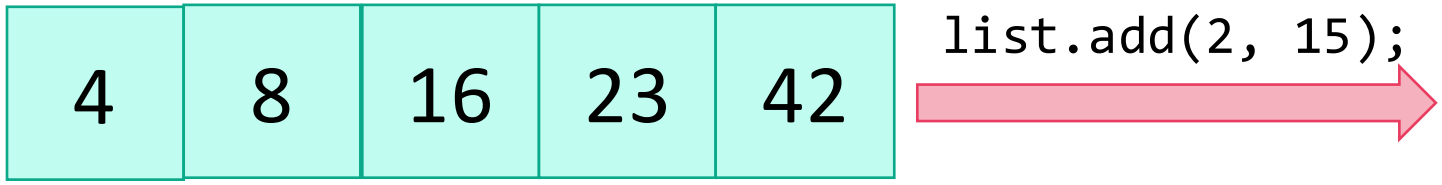

```
list.size(): 5
```

ArrayList (3/4)

ArrayLists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - ArrayLists can only hold Objects, so might need to use "wrapper" types: `Integer, Double, Boolean, Character, etc.`

But ArrayLists have dynamic length (so they can resize!)



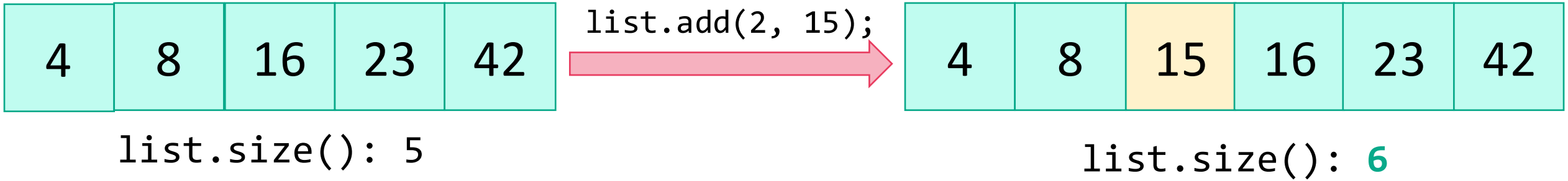
`list.size(): 5`

ArrayList (4/4)

ArrayLists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - ArrayLists can only hold Objects, so might need to use "wrapper" types: `Integer, Double, Boolean, Character, etc.`

But ArrayLists have dynamic length (so they can resize!)



ArrayList Methods

Method	Description
`add(type element)`	Adds *element* to the ***end*** of the `ArrayList`
`add(int index, type element)`	Adds *element* to the specified ***index*** in the `ArrayList`
`size()`	Returns the number of elements in the `ArrayList`
`contains(type element)`	Returns `true` if *element* is contained in the `ArrayList`, `false` otherwise
`get(int index)`	Returns the element at *index* in the `ArrayList`
`remove(int index)`	Removes the element at *index* from the `ArrayList` and returns the removed element.
`indexOf(type element)`	Returns the index of *element* in the `ArrayList`; returns -1 if the *element* doesn't exist in the `ArrayList`
`set(int index, type element)`	Sets the element at *index* to the given *element* and returns the old value

ArrayList Methods Usage (1/2)

- Whenever referring to “the ArrayList”, we are referring to the ArrayList we’re calling the method on!

```
List<String> list = new ArrayList<String>();
list.add("hello");
list.add(0, "world");
list.indexOf("world"); // what is the output?
```

ArrayList Methods Usage (2/2)

- Whenever referring to "the ArrayList", we are referring to the ArrayList we're calling the method on!

```
List<String> list = new ArrayList<String>();
list.add("hello");
list.add(0, "world");
list.indexOf("world"); // what is the output?
```

```
String[] list = new String[2];
list[0] = "hello";
list[0] = "world";
list[1] = "hello";
//... indexOf?
```

Lecture Outline (5/5)

- Announcements
- Movie Ratings
- Using `PrintStream` for File Output
- ArrayList Recap
- **ArrayList Examples**

loadFromFile

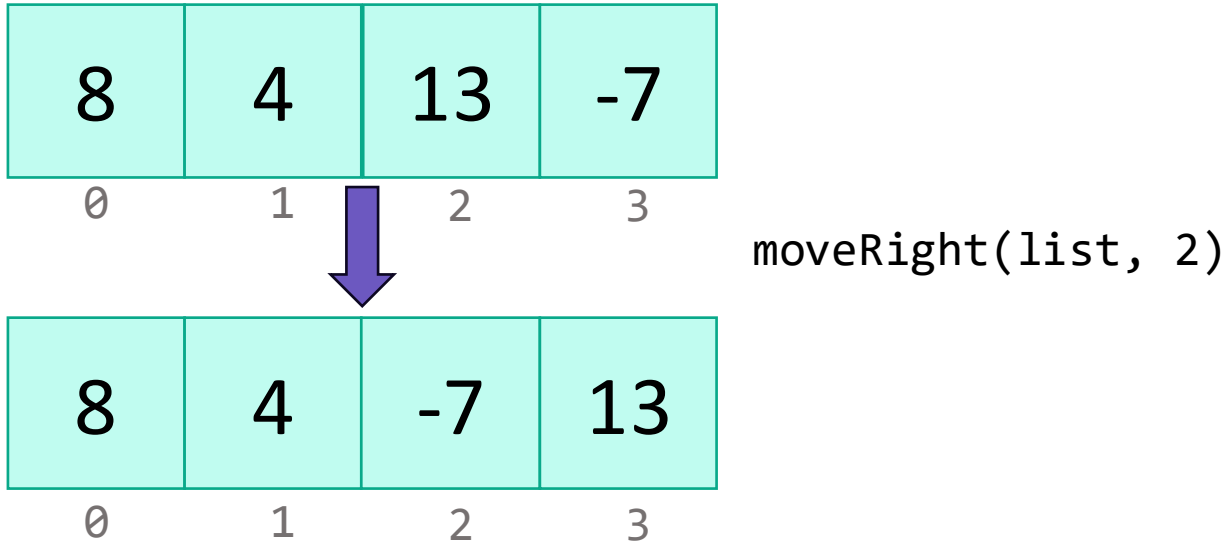
Write a method called `loadFromFile` that accepts a `Scanner` as a parameter and returns a new `ArrayList` of `Strings` where each element of the `ArrayList` is a line from the `Scanner`, matching the order of the `Scanner`'s contents.

e.g., the first line in the `Scanner` is stored at index `0`, the next line is stored at index 1, etc.

moveRight

Write a method called `moveRight` that accepts an `ArrayList` of integers `list` and an `int` `n` and moves the element at index `n` one space to the right in `list`.

For example, if `list` contains `[8, 4, 13, -7]` and our method is called with `moveRight(list, 2)`, after the method call `list` would contain `[8, 4, -7, 13]`.



Practice : Think

sli.do #cse122

What ArrayList methods (and in what order) could we use to implement the `moveRight` method?

A)
```
list.remove(n);
list.add(n);
```

B)
```
int element = list.remove(n);
list.add(n, element);
```

C)
```
list.add(n);
list.remove(n-1);
```

D)
```
int element = list.remove(n);
list.add(n+1, element);
```

Practice : Pair

sli.do #cse122

What ArrayList methods (and in what order) could we use to implement the `moveRight` method?

A)
```
list.remove(n);
list.add(n);
```

B)
```
int element = list.remove(n);
list.add(n, element);
```

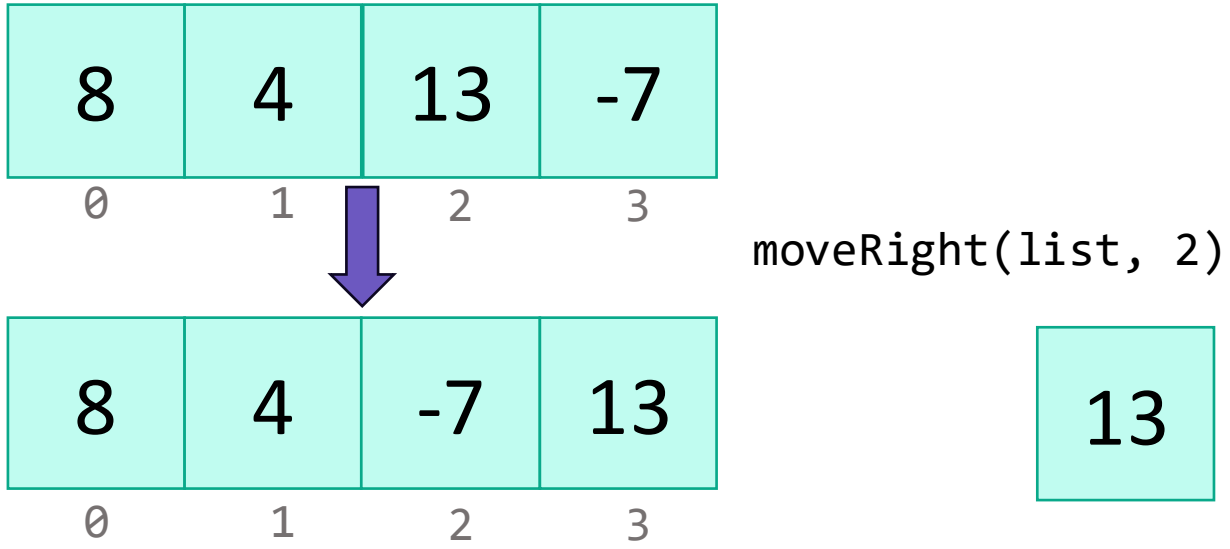
C)
```
list.add(n);
list.remove(n-1);
```

D)
```
int element = list.remove(n);
list.add(n+1, element);
```

moveRight

Write a method called `moveRight` that accepts a `List` of integers `list` and an `int` n and moves the element at index n one space to the right in `list`.

For example, if `list` contains `[8, 4, 13, -7]` and our method is called with `moveRight(list, 2)`, after the method call `list` would contain `[8, 4, -7, 13]`.



Edge Cases! (And Testing)

When writing a method, especially one that takes input of some kind (e.g., parameters, user input, a Scanner with input) it's good to think carefully about what assumptions you can make (or cannot make) about this input.

Edge case: A scenario that is uncommon but possible, especially at the "edge" of a parameter's valid range.

❓ What happens if the user passes a negative number to `moveDown`?

❓ What happens if the user passes a number larger than the length of the list to `moveDown`?

More testing tips on the course website's Resources page!

compareToList

Write a method called `compareToList` that accepts two `ArrayLists` of integers `list1` and `list2` as parameters and compares the elements of the two lists, printing out the locations of common elements in each of the `ArrayLists`.

For example, if `list1` contained `[5, 6, 7, 8]` and `list2` contained `[7, 5, 9, 0, 2]`, a call to `compareToList(list1, list2)` would produce output such as:

```
- 5 (list1 at 0, list2 at 1)
- 7 (list1 at 2, list2 at 0)
```

ArrayList Methods

Method	Description
`add(type element)`	Adds *element* to the ***end*** of the `ArrayList`
`add(int index, type element)`	Adds *element* to the specified ***index*** in the `ArrayList`
`size()`	Returns the number of elements in the `ArrayList`
`contains(type element)`	Returns `true` if *element* is contained in the `ArrayList`, `false` otherwise
`get(int index)`	Returns the element at *index* in the `ArrayList`
`remove(int index)`	Removes the element at *index* from the `ArrayList` and returns the removed element.
`indexOf(type element)`	Returns the index of *element* in the `ArrayList`; returns -1 if the *element* doesn't exist in the `ArrayList`
`set(int index, type element)`	Sets the element at *index* to the given *element* and returns the old value

topN

Write a method called `topN` that accepts an `ArrayList` of characters `list` and an `int` n and returns a new `ArrayList` of characters that contains the first n elements of list.

For example, if `list` contained
`['m', 'a', 't', 'i', 'l', 'd', 'a']`,
a call to `topN(list, 4)` would return an `ArrayList` containing `[‘m', ‘a', ‘t', ‘i']`