

LEC 05

ArrayList Applications

Questions during Class?

Raise hand or send here

sli.do #cse122



BEFORE WE START


*Talk to your neighbors:
Any weekend plans?*

Music: [122 25wi Lecture Tunes](#) 

Instructor: Elba Garza

TAs:	Anya	Daniel Ryan	Ken	Nicole
	Ashley	Diya	Kuhu	Nicole
	Cady	Elizabeth	Kyle	Niyati
	Caleb	Hannah	Leo	Sai
	Carson	Harshitha	Logan	Steven
	Chaafen	Ivory	Maggie	Yang
	Colin	Izak	Mahima	Zach
	Connor	Jack	Marcus	
	Dalton	Jacob	Minh	

Lecture Outline

- **Announcements** 
- Warm Up
- ArrayList Extended Application

Announcements

- C0 grades and R0 out yesterday
 - Now that you have your first set of grades, review the [Course Grades](#) section of the syllabus to understand how they factor into your grade at the end of the quarter!
 - [Grade Checker spreadsheet](#) also linked from the syllabus to help track your grades throughout the quarter.
 - See [Resubmission page](#) and [Ed post](#) for R0 logistics
- Creative Assignment 1 (C1) out later today!
 - Focused on ArrayLists
 - Due next Thursday, Jan 30th by 11:59 PM
- First quiz in section on Tuesday Jan 28th
 - Practice Quiz 0 will be released later tonight!

Lecture Outline

- Announcements
- **Warm Up** 
- ArrayList Extended Application

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!

ArrayList Methods

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

addAll

Write a method called `addAll` that accepts two `ArrayLists` of `Characters`, `list1` and `list2`, and an integer `location` as parameters and inserts all of the elements from `list2` into `list1` at the specified `location`.

Lecture Outline

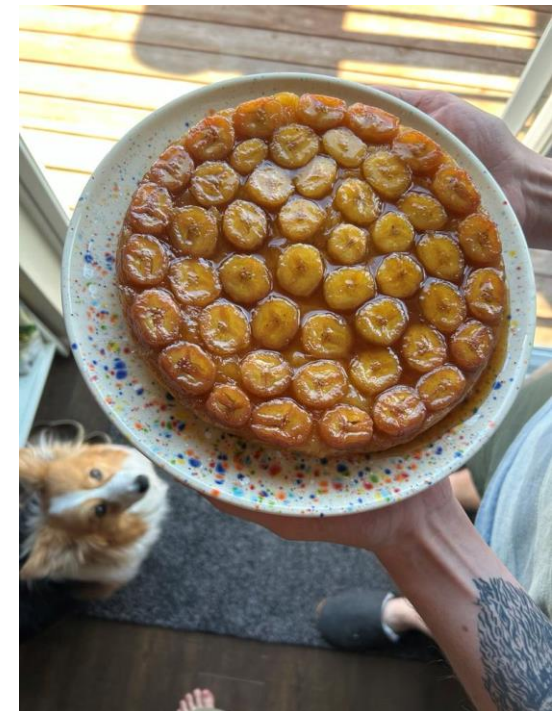
- Announcements
- Warm Up
- **ArrayList Extended Application** ◀

Bakery Favorites

We will write a program called `BakeryFavorites.java` that manages a list of favorite bakeries for a user (using an `ArrayList`) and allows the user to perform various different operations on their stored list of favorite bakeries.

Key skills used:

- User Interaction (UI) loop
- Iterative development strategies
- Functional decomposition
- Practice with `ArrayList` methods!



Bakery Favorites: Operations

- Load a list of favorites in from a file provided by the user.
- Compare the stored list of favorites to another list of favorites provided by the user in another file.
- Report the top n favorites according to the list, where the user can specify n .
- Move a specific favorite down in the list.
- Add a list of favorites in a user-provided file to the stored list of favorites at a specified location.
- Save the current list of favorites to a file provided by the user.

Bakery Favorites: Development Strategy

- Set up the main scaffold code
- Menu loop
- Develop each operation, one at a time

You'll see a similar development strategy in Creative Project 1's specification — we recommend you follow it!

Bakery Favorites: Operations

- Load a list of favorites in from a file provided by the user.
- Compare the stored list of favorites to the list of favorites provided by the user.
- Report the top n favorites in the list, where the user can specify n .
- Move a specific favorite down in the list.
- Add a list of favorites in a user-provided file to the stored list of favorites at a specified location.
- Save the current list of favorites to a file provided by the user.

ALREADY DONE!
See In-Class 4