

LEC 15

CSE 122

Collections

BEFORE WE START

*Talk to your neighbors:**coffee or tea?**Music: [Miya's 23wi CSE 122 Playlist](#)*Instructor **Miya Natsuhara**

TAS

Ayush	Atharva	Ernie	Ambika
Connor	Julia	Di	Elizabeth
Poojitha	Megana	Logan	Joe
Andrew A	Joey	Shivani	Jin
Andrew C	Eesha	Michelle	Ben
Jasmine	Lilian	Steven	Evelyn
Darel	Thomas	Kevin	Kent
Gabe	Leon	Ken	
Karen	Melissa	Vivek	
Colton	Audrey	Autumn	


Questions during Class?

Raise hand or send here

sli.do #cse122



# Lecture Outline

- **Announcements** 
- Optional
- Recap of Collections
- Dumb Data Structures
- Collections

# Announcements

- P3 will be released later today!
  - OOP, Interfaces
- Quiz 2 details posted
  - Take home
  - No enforced time limit
  - Collaboration permitted *with caution* ⚠
- Final Exam info posted

# Lecture Outline

- Announcements
- **Optional** ◀
- Recap of Collections
- Dumb Data Structures
- Collections

# (PCM) Optional

`Optional` is a Java class that is used to handle situations where a value is *sometimes* there.

You give `Optional` a type to hold (or potentially not hold) when you are referring to its type.

e.g., `Optional<String>`, `Optional<Integer>`, `Optional<Point>`

# (PCM) Optional Methods

Method	Description
<code>Optional.empty()</code>	Creates an empty <code>Optional</code> object
<code>Optional.of(...)</code>	Creates an <code>Optional</code> object holding the object it's given
<code>isEmpty()</code>	Returns <code>true</code> if there <i>is no</i> value stored, and <code>false</code> otherwise
<code>isPresent()</code>	Returns <code>true</code> if there <i>is a</i> value stored, and <code>false</code> otherwise
<code>get()</code>	Returns the stored object from the <code>Optional</code> (if one is stored; otherwise throws a <code>NoSuchElementException</code> )

The `Optional` class has more than just these methods, but these are what you'll need to focus on for this class!

## (PCM) Optional Methods

`isEmpty()`, `isPresent()`, and `get()` are called like normal instance methods (on an actual instance of `Optional`).

`Optional.of(...)` and `Optional.empty()` are called differently

(Like the `Math` class methods)

## (PCM) Why Optional?

Using `Optional` can help programmers avoid `NullPointerException`s by making it explicit when a variable may or may not contain a value.

- Remember – `null` refers to the absence of an object!

There are other `Optional` methods (that you should explore in your own time if you're interested) that can be really useful to cleanly work with data that may or may not be present.



# Student / Course Example one more time...

Let's add two more methods to `Course.java`:


```
public void setCourseEvalLink(String url)
```

```
public Optional<String> getCourseEvalLink()
```

The link to the evaluations for a course doesn't usually exist until the last few weeks of the quarter. What if a client calls `getCourseEvalLink` before one is set up?

`Optional` to the rescue!

# Lecture Outline

- Announcements
- Optional
- **Recap of Collections** 
- Dumb Data Structures
- Collections

# Goal for Today


Review some of the data structures we've talked about this quarter

Understand how Java organizes them with *interfaces*

# Collections: What *classes* have we seen so far?

# Collections: What *interfaces* have we seen so far?

# Lecture Outline

- Announcements
- Optional
- Recap of Collections
- **Dumb Data Structures** 
- Collections

# Dumb Data Structures

We're going to create our own versions of these classes so we can dig into how they all relate to each other!

BUT they're going to be real dumb.

If you want to get a sense of how they're *actually* implemented, go take CSE 123!

# DumbArrayList

<code>DumbArrayList()</code>	<code>set(int index, int value)</code>
<code>add(int value)</code>	<code>remove(int index)</code>
<code>add(int index, int value)</code>	<code>size()</code>
<code>contains(int value)</code>	<code>indexOf(int value)</code>
<code>isEmpty()</code>	<code>toString()</code>
<code>get(int index)</code>	



# Lecture Outline

- Announcements
- Optional
- Recap of Collections
- Dumb Data Structures
- **Collections** ◀

# DumbIntCollection Relationships