

LEC 06

CSE 122

Sets; for-each; Iterators

Questions during Class?
Raise hand or send here

sli.do #cse122



BEFORE WE START

Talk to your neighbors:

*What is the last live entertainment
(e.g. concert, play, sporting event)
you attended?*

Respond on sli.do!

Instructor: Brett Wortzman

TAs:	Archit	Daniel	Naomi	Sreshta
	Blake P.	David	Neha	Suyash
	Blake R.	Hanna	Nicole	Trisha
	Cady	Isis	Rio	Wesley
	Carson	Jun	Rohan	Yash
	Cole	Landon	Shreyank	
	Connor	Michelle	Somya	

Music:  [CSE 122 26au Lecture Tunes](#) 

Lecture Outline

- **Announcements** 
- Review: Sets
- Programming Practice
- for-each Loop
- Iterators

Announcements

- CQ0 ends on Friday
 - *Zero* remaining open slots
 - If you haven't scheduled yet, contact Brett **immediately** to discuss options
- P1 out Friday, Oct. 23; due Thursday, Oct. 29
- PQ0 on Tuesday, Oct. 27 in section
 - [Policies posted on Ed](#)
 - Will cover material through Lesson 5 (up to stacks)

Lecture Outline

- Announcements
- **Review: Sets** 
- Programming Practice
- for-each Loop
- Iterators

(PCM) Sets (ADT)

- An *unordered* collection of *unique* values that can perform the following operations efficiently:
 - add
 - remove
 - search (contains)
- Sets *do not have indices*!



"hi" "hola"
"hallo" "hello"
"tervetuloo"

(PCM) Sets in Java

- Set is an interface in Java
 - In `java.util` package, like `List`, `Queue`, and `Stack`
- Two implementations: `HashSet` and `TreeSet`
 - `HashSet`: Very fast! Implemented using a *hash table*
 - *Elements are stored in an unpredictable order*
 - `TreeSet`: Pretty fast! Implemented using a *binary search tree*
 - *Elements are stored in sorted order*

(PCM) Set Methods

Method	Description
<code>add(value)</code>	Adds the given value to the set, returns whether or not the given value was added successfully
<code>contains(value)</code>	Returns <code>true</code> if the given value is found in this set
<code>remove(value)</code>	Removes the given value from the set; returns <code>true</code> if the set contained the value, <code>false</code> if not
<code>clear()</code>	Removes all elements from the set
<code>size()</code>	Returns the number of elements in list
<code>isEmpty()</code>	Returns <code>true</code> if the set's size is 0; <code>false</code> otherwise
<code>toString()</code>	Returns a <code>String</code> representation of the set such as <code>"[3, 42, -7, 15]"</code>

Lecture Outline

- Announcements
- Review: Sets
- **Programming Practice** 
- for-each Loop
- Iterators

Lecture Outline

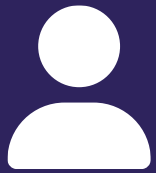
- Announcements
- Review: Sets
- Programming Practice
- **for-each Loop** ◀
- Iterators

(PCM) for-each Loop

- A new kind of loop!

```
Set<String> words = new HashSet<>();  
for (String s : words) {  
    System.out.println(s);  
}
```

- Works for almost any collection (including arrays!)



Practice : Think



sli.do

#cse122

What output is produced by this code?

```
Set<Integer> nums = new TreeSet<>();  
nums.add(3);  
nums.add(9);  
nums.add(3);  
nums.add(-2);  
nums.add(0);
```

```
for (int n : nums) {  
    System.out.print(n + " ");  
}
```

A. 3 9 3 -2 0

B. -2 0 3 3 9

C. -2 0 3 9

D. 9 3 0 -2

E. ConcurrentModificationException



Practice : Pair



sli.do

#cse122

What output is produced by this code?

```
Set<Integer> nums = new TreeSet<>();  
nums.add(3);  
nums.add(9);  
nums.add(3);  
nums.add(-2);  
nums.add(0);
```

```
for (int n : nums) {  
    System.out.print(n + " ");  
}
```

A. 3 9 3 -2 0

B. -2 0 3 3 9

C. -2 0 3 9

D. 9 3 0 -2

E. ConcurrentModificationException

Lecture Outline

- Announcements
- Sets Review
- Programming Practice
- for-each Loop
- **Iterators** 

Iterators

A new object that has access to all of the elements of a given structure and can give them to you, one at a time.

Iterators

- Returned by the `iterator()` method

Methods	Description
<code>hasNext()</code>	Returns true if there are more elements for the iterator to return
<code>next()</code>	Returns the next element in the iteration
<code>remove()</code>	Removes and returns the element that was last returned by <code>next()</code>

- You must use the iterator's `remove()` method to remove things from what you're iterating over – otherwise you will get a **ConcurrentModificationException**