

LEC 03

CSE 122

Lists

Questions during Class?  
Raise hand or send here

sli.do #cse122



BEFORE WE START

*Talk to your neighbors:**How do you like to spend a rainy day?*

Respond on sli.do!

**Instructor:** Brett Wortzman

| <b>TAs:</b> | 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/Reminders** ◀
- Using `PrintStream` for File Output
- PCM Review
- Programming Practice

# Announcements

- Programming Project 0 (P0) out later today
  - Due next Thursday, October 15<sup>th</sup>
  - Focused on File I/O and Lists
- Computer Quiz 0 will run Friday 10/16 through Friday 10/23
  - More details on scheduling and logistics coming next week
  - Computer quizzes will be Bring Your Own Device; contact us ASAP if you don't have a suitable device (laptop or tablet w/keyboard)
- [IPL schedule and instructions](#) posted on website

# Participation Grade

- Two ways to earn credit:
  - Attempt every slide in a PCM by 11:59pm the day of class
    - “Attempt” means at least one submission or read/view the reading/video
    - Can choose just one of the reading/video pair
  - Attend and participate in section
    - Paper quiz days not included
    - “Participate” means any sort of active engagement as evaluated by your TA
- There will be 32 total opportunities (16 PCMs, 16 sections)
  - Earn credit for 26/32 for an E; 22/32 for an S
  - Section 0 and PCM 1 were freebies for everybody
- Make-ups will generally not be granted
  - But you can miss six and still get an E

See [description](#) and [grading](#) in the syllabus

# Lecture Outline

- Announcements/Reminders
- Using `PrintStream` for File Output ◀
- PCM Review
- Programming Practice

# (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                                                                      |
|---------------------------|----------------------------------------------------------------------------------|
| <code>print(...)</code>   | Prints the given value to the set output location.                               |
| <code>println(...)</code> | 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!

# Lecture Outline

- Announcements/Reminders
- Using `PrintStream` for File Output
- **PCM Review** ◀
- Programming Practice

# (PCM) Lists / ArrayList

Lists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - In Java, ArrayList can only hold objects, so might need to use “wrapper” types: Integer, Double, Boolean, Character, etc.

**But** Lists have dynamic length (so they can resize!)

# (PCM) Lists / ArrayList

Lists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - In Java, ArrayList can only hold objects, so might need to use “wrapper” types: Integer, Double, Boolean, Character, etc.

**But** Lists have dynamic length (so they can resize!)

|   |   |    |    |    |
|---|---|----|----|----|
| 4 | 8 | 16 | 23 | 42 |
|---|---|----|----|----|

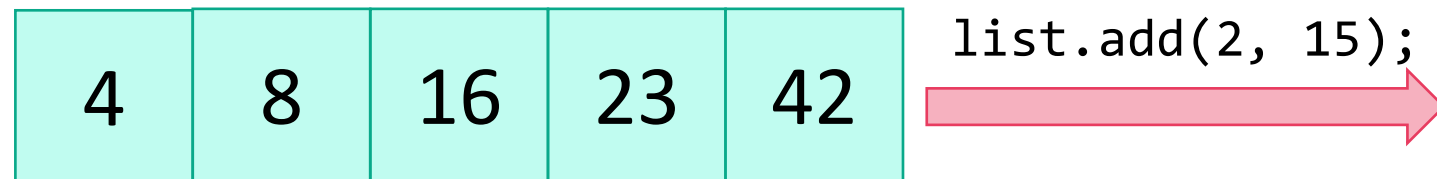
```
list.size(): 5
```

# (PCM) Lists / ArrayList

Lists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - In Java, ArrayList can only hold objects, so might need to use “wrapper” types: Integer, Double, Boolean, Character, etc.

**But** Lists have dynamic length (so they can resize!)



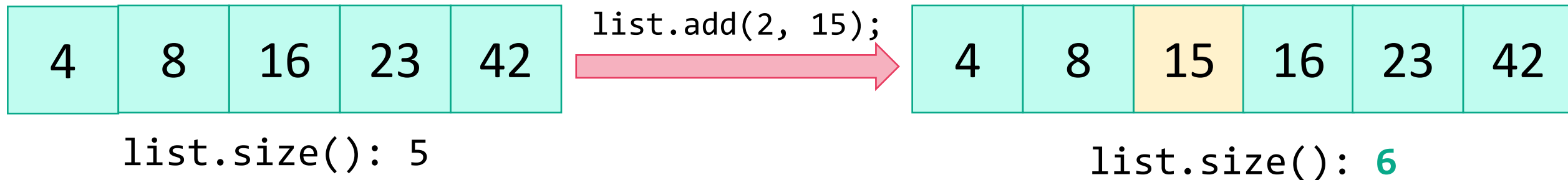
`list.size(): 5`

# (PCM) Lists / ArrayList

Lists are very similar to arrays

- Can hold multiple pieces of data (elements)
- Zero-based indexing
- Elements must all have the same type
  - In Java, ArrayList can only hold objects, so might need to use “wrapper” types: Integer, Double, Boolean, Character, etc.

**But** Lists have dynamic length (so they can resize!)



# (PCM) ArrayList Methods

| Method                                                  | Description                                                                                                           |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>add(type <i>element</i>)</code>                   | Adds <i>element</i> to the <i>end</i> of the ArrayList                                                                |
| <code>add(int <i>index</i>, type <i>element</i>)</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 <i>element</i>)</code>              | Returns true if <i>element</i> is contained in the ArrayList, false otherwise                                         |
| <code>get(int <i>index</i>)</code>                      | Returns the element at <i>index</i> in the ArrayList                                                                  |
| <code>remove(int <i>index</i>)</code>                   | Removes the element at <i>index</i> from the ArrayList and returns the removed element.                               |
| <code>indexOf(type <i>element</i>)</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 <i>index</i>, type <i>element</i>)</code> | Sets the element at <i>index</i> to the given <i>element</i> and returns the old value                                |

# ArrayList Methods Usage

- 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

- 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

- Announcements/Reminders
- Using `PrintStream` for File Output
- PCM Review
- **Programming Practice** ◀

# Programming Practice

- Four tasks:
  - `readAllLines` takes a `Scanner` as a parameter and returns a list containing all the lines read from that `Scanner` in order
  - `getWords` takes a `String` as a parameter and returns a list containing all the words (tokens) in that `String` in order
  - `getWordsReverse` takes a `String` as a parameter and returns a list containing all the words (tokens) in that `String` in **reverse** order
  - `pullLongWords` takes a list of `Strings` as a parameter, removes all `Strings` longer than five characters from the list, and returns a new list containing the removed `Strings` in the order they appeared in the original list.