Readings
This Week: Review!
(Reminder: Readings are absolutely vital for learning this stuff!)

Labs and Tutorials
No labs/tutorials this week, during week of midterm.

Midterm – Tonight!
- Midterm #2 is 6-7pm on March 11 (Wednesday) in Woodward IRC 2
- Midterm coverage is through Ch 7.

Learning Goals
By the end of class today you will be able to...
- Write programs with multi-dimensional arrays, arrays of objects, and arrays inside objects.
- Read and write programs that use partially-filled arrays (in the typical, idiomatic manner).
- Read and write programs that use Java’s ArrayList class.
- Implement your own ArrayList class.
- (Ulterior Learning Goal: Get increasingly comfortable with arrays…)

Last Time: Organizing Data
- Create a CPSC111Student class!
  - Constructors
  - Methods to get and set name
  - Methods to get and set lab scores.
  - Method to compute total lab score.
  - Etc.
- For simplicity, let’s just worry about the student’s name and lab scores…
Organizing Data

- Create a CPSC111Student class!
- Constructors
  - Create a new student, specify number of labs.
  - Create a new student, specify name and number of labs.
- Methods to get and set name
- Methods to get and set lab scores.
- Method to compute total lab score.
- Etc.

For simplicity, let's just worry about the student's name and lab scores…

getTotalLabScore() Version 1

- We can add up the lab scores each time called.
- This can be wasteful if getTotalLabScore is called often – the same work gets re-done repeatedly.
**getTotalLabScore() Version 2**
- We can put the result in a variable `totalLabScore`.
- `getTotalLabScore()` just returns that value.
- No wasted computation when `getTotalLabScore()` called.

**getTotalLabScore() Version 2**
- We can put the result in a variable `totalLabScore`.
- `getTotalLabScore()` just returns that value.
- No wasted computation when `getTotalLabScore()` called.
- But when does `totalLabScore` get computed?

**getTotalLabScore() Version 3**
- We can put the result in a variable `totalLabScore`.
- `getTotalLabScore()` just returns that value.
- We add a new method `computeTotalLabScore()` that computes `totalLabScore`.

**getTotalLabScore() Version 3**
- We can put the result in a variable `totalLabScore`.
- `getTotalLabScore()` just returns that value.
- We add a new method `computeTotalLabScore()` that computes `totalLabScore`.
- Gives correct results, but only if user always remembers to call `computeTotalLabScore()`.
- Bad Design: Violates data protection!

**getTotalLabScore() Version 4**
- We can put the result in a variable `totalLabScore`.
- `getTotalLabScore()` just returns that value.
- The object keeps track of whether `totalLabScore` is up-to-date. When `getTotalLabScore()` is called, it calls `computeTotalLabScore()` only if needed.

**Questions?**
Teen Talk Barbie™ Reloaded
- A few lectures back, we wrote a program that learns some phrases and prints them back at random (inspired by Mattel's Teen Talk Barbie™ doll).
- The goal was to have a fun example to introduce arrays.
- But the programming style wasn't very good.
- Better to create a TalkingDoll class...

TalkingDoll Class
- Constructor that specifies name and maximum vocabulary size.
- Getters/Setters for the name.
- A method to add phrases to the doll.
- A method to get a random phrase back.

Partially Filled Arrays
- Arrays have a fixed size, but often we want to hold a variable number of items.
- A Very Common Solution:
  - Make an array bigger than you need.
  - Have an int variable to keep track of what you are actually using.

Partially Filled Arrays
- Arrays have a fixed size, but often we want to hold a variable number of items.
- A Very Common Solution:
  - Make an array bigger than you need.
  - Have an int variable to keep track of what you are actually using.

Partially Filled Arrays
- Arrays have a fixed size, but often we want to hold a variable number of items.
- A Very Common Solution:
  - Make an array bigger than you need.
  - Have an int variable to keep track of what you are actually using.

Partially Filled Arrays
- A Very Common Solution:
  - Your code must keep count variable up-to-date!
  - Good to keep these private, so your methods can always do the right thing.
  - (This is an example of a class or object invariant.)
Java’s ArrayList Class

- What if you don’t know (or want to force) a fixed maximum size in advance?
- Java has an ArrayList class for this case:
  - import java.util.ArrayList
  - Declare:
    ```java
    ArrayList<String> phrases = new ArrayList<String>();
    ```

Methods:
- int size()
- void add(<type> newValue)
- <type> get(int index)
- void set(int index, <type> newValue)

- <type> can be any object type. See Ch 7.2.

TalkingDoll with ArrayLists

- Let’s redo the TalkingDoll class using ArrayLists…

Do-It-Yourself ArrayLists

- ArrayLists are nothing magical!
  - (OK, the generic <type> stuff is kind of magic.)
  - It’s just a class. If we fix the type of the elements (e.g., have an ArrayList of String), you know enough to write your own version.
  - But how do you allow arrays to grow?

Real-Life Analogy: Moving Homes

- A house (or condo, apartment, etc.) has a fixed size. What happens when your family grows and you need more space?

Real-Life Analogy: Moving Homes

- A house (or condo, apartment, etc.) has a fixed size. What happens when your family grows and you need more space?
- Answer: You buy a bigger place, and then you pack up and move all your stuff to the new place, and get rid of your old home.
Making Your Own ArrayList

An array has a fixed size. What happens when your list grows and you need more space?

Answer: You allocate a bigger array, and then you pack up and move all your stuff to the new array, and get rid of your old array.

```
Making Your Own ArrayList

Answer: You allocate a bigger array, and then you pack up and move all your stuff to the new array, and get rid of your old array.
```

```
Making Your Own ArrayList

Answer: You allocate a bigger array, and then you pack up and move all your stuff to the new array, and get rid of your old array.
```

```
Making Your Own ArrayList

Answer: You allocate a bigger array, and then you pack up and move all your stuff to the new array, and get rid of your old array.
```

```
Making Your Own ArrayList

Answer: You allocate a bigger array, and then you pack up and move all your stuff to the new array, and get rid of your old array.
```
**Making Your Own ArrayList**

- Answer: You allocate a bigger array, and then you pack up and move all your stuff to the new array, and get rid of your old array.

<table>
<thead>
<tr>
<th>aCount</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>3 1 4 1</td>
</tr>
<tr>
<td>a.length</td>
<td>4</td>
</tr>
<tr>
<td>newA</td>
<td>3 1 4 1</td>
</tr>
<tr>
<td>newA.length</td>
<td>8</td>
</tr>
</tbody>
</table>

**MyStringArrayList**

- Let's create a MyStringArrayList class.
- Methods:
  - `int size()`
  - `void add(String newString)`
  - `String get(int index)`