Android Basics: Multiscreen Apps

Arrays, Lists, Loops & Custom Classes

- How to some a list of words in the app (Arrays, ArrayLists)
- Display a list of words
- Display a list of English/Miwok word pairs
- Add the words from all the remaining categories

Arrays

All variables need a data type.

You can have an array of any data type:

- int
- double
- boolean
- char
- long
- float
- short
- byte
- Objects

```
// Create array
int [] showSizesAvailable = new int[3];

// Initialize elements in an array
shoeSizesAvailable[0] = 5;
shoeSizesAvailable[1] = 7;
shoeSizesAvailable[2] = 10;

// Access elements in an array
shoeSizesAvailable[0] = 5;
shoeSizesAvailable[1] = 7;
shoeSizesAvailable[2] = 10;

// Get the array length
shoeSizesAvailable.length // -> value of 3
```

Array vs ArrayList

	Array	ArrayList
Is it a class?	No	Yes
Can it change size once created?	No	Yes
Does it use methods to access and modify elements?	No	Yes

ArrayLists

1. Documentation: List | Android Developers (https://developer.android.com/reference/java/util/List.html)
2. Documentation: AbstractList | Android Developers (https://developer.android.com/reference/java/util/AbstractList.html)
3. Documentation: ArrayList | Android Developers (https://developer.android.com/reference/java/util/ArrayList.html)

Concrete Class: ArrayList

```
public class ArrayList {

    int size;
    Object[] elements;

    boolean add(E e) {
        elements[size++] = e;
    }

    E remove(int index){
        elements[--size] = null;
    }

    E get(int index) {
        return(E) elements[index];
    }

...
}
```

Abstract Class: AbstractList

```
public abstract class AbstractList {

    boolean add(E e) {
        add(size(), e);
        return true;
}

    E get(int index);

...
}
```

Interface: List

```
// No implementation of its methods
public interface List {

    boolean add(E object);

    E remove(int index);

    E get(int index);
}
```

Type Parameters

Symbol	Type
E	Element
K	Key
N	Number
T	Type
V	Value
S, U, V, etc.	2nd, 3rd, 4th types (when there is more than one parameter)

How to Create and Access Elements in an ArrayList

Create an ArrayList

```
ArrayList<String> musicLibrary = new ArrayList<String>();
```

Add elements in an ArrayList

```
musicLibrary.add("Yellow Submarine");

musicLibrary.add("Thriller");

// Adds an element at a specific index

musicLibrary.add(0, "Blue Suede Shoes");
```

Access elements in an ArrayList

```
musicLibrary.get(0);

musicLibrary.get(1);

musicLibrary.get(2);
```

Remove elements from an ArrayList

```
// Remove the element at the specific index

musicLibrary.remove(2);
```

Get ArrayList length or size

```
musicLibrary.size();
```

Loops

While Loops

The code within the loop will be executed until the condition in parentheses is `false`

```
while(<condition>){
    <instructions>
}
```

Steps:
1. What is the task to repeat?
->
2. How many times to repeat?
3. What is the condition?
-> Count variable < 3

```
int count = 0;

while(count < 3){
    playSound();
    count += 1;
}
```

Counter Shorthands

Shorthand	Meaning
i++	i = i + 1
j--	j = j - 1
j += 3	j = j +3

For Loop

```
for (int index=0; index < 3; index++){
    Log.v("NumbersActivity", "Index: " + index + "Value: " + words.get(index));
}
```

View Recycling

When looking at 1000 contacts,
Only 5 will appear on the screen at any time.
Instead of creating 1000 TextViews, we create 5, and change the content as the user scrolls.
This is called ~view recycling~.

Using a ListView and an ArrayAdapter, we can create just enough views based on what we need to fill the screen.

We create a bunch of views, and have a pile of ~scrap views~, that aren't currently being shown on the screen.

When a user scrolls down, and a View goes off of the top of the screen, it gets cycled back around to the top of the screen to be used again.

When a user scrolls up, and a View goes off the bottom of the screen, it gets cycled back around to the bottom of the screen to be used again.

DEPRECATED: ~~Documentation: Memory Monitor | Android Studio (https://developer.android.com/studio/profile/am-memory.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)~~

~Memory Monitor~ : a feature in Android Studio that keeps track of your application's memory usage (DEPRECATED)

Documentation: View the Java Heap and Memory Allocations with Memory Profiler | Android Studio (https://developer.android.com/studio/profile/memory-profiler.html)
~Memory Profiler~ : a feature introduced in Android Studio 3.0 that helps you identify memory leaks and memory churn.

programming/java/android/beginner/3-multiscreen-apps/2-arrays-lists-loops-custom-classes