1. Log in

2. Working in Unit 3
   - 1-Dimensional Arrays
   - 2-Dimensional Arrays
   - Array Lists

Oct 28-7:39 PM

Arrays - int[ ] theArrayName = new int[4];

* Defined array length ... could not change.
* Array may or may not be "full"
* We had to manually shift #'s/sort in an array.
* If partially full, we have to track "how full it is"

ArrayList ArrayList<Type> theListName = new ArrayList(size);

* Notice there is no [ ] used
* ArrayLists are "fluid" in size = adjustable
* We can add and subtract elements anywhere!
* We don't have to manually move elements, it is done automatically for us!!!

Oct 28-7:41 PM

Lets create our first arrayList - **ourClassmates** ...

Do you remember ...

Scanner Object needs import.java.util.*
DecimalFormat Object needs import.java.text.*

Today ...

ArrayList Object needs import.java.util.*

Oct 28-7:52 PM

Do you remember ...

```
Scanner theScannerName = new Scanner(System.in);

DecimalFormat it = new DecimalFormat("##.00");
```

This isn't any different ...

```
ArrayList<String> ourClassMates = new ArrayList(20);
```

Oct 31-9:06 AM

Create the List! It has 20 elements!

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

ourClassMates.add("Mr. Henderson");      //1st, index 0
ourClassMates.add("Justin");             //2nd, index 1
ourClassMates.add("Hannah");             //3rd, index 2
```

listName.add(x) ... This will add the 'x' item on the end of the list!

Oct 31-9:12 AM

Let's check our list by printing it out ...

```
for(int i=0;i<ourClassMates.size();i++)
    System.out.println("Index "+i+" is "+ourClassMates.get(i));
```

*** Do you understand the above code??? ***

Oct 31-9:23 AM

ArrayList Methods!

Do you remember ...

listName.add(x)	Adds item to end of list

Here's more ...

listName.add(#, x)	Add x at index #, shift
listName.remove(#)	Remove item at index #
listName.get(#)	Get item at index #
listName.indexOf(x)	Get index # of an item
listName.size( )	Returns size of list
listName.set(#,x)	Set index # to item x

Oct 28-7:52 PM

An Example of Each ...

```
ourClassMates.add("Mr. Henderson");
ourClassMates.add("Justin");
ourClassMates.add("Hannah");
```

```
ourClassMates.add(1,"Monster");
```

Now, the list has changed to contain 4 items, in this order:

Mr. Henderson
Monster
Justin
Hannah

Oct 28-7:52 PM

An Example of Each ...

```
ourClassMates.add("Mr. Henderson");
ourClassMates.add("Justin");
ourClassMates.add("Hannah");
```

```
ourClassMates.remove("Mr. Henderson");
```

Now, the list has been shortened!
The list now only has 2 elements, Justin is now index 0!

We also could have used ...

```
ourClassMates.remove(0);
```

Oct 28-7:52 PM

An Example of Each ...

```
ourClassMates.add("Mr. Henderson");
ourClassMates.add("Justin");
ourClassMates.add("Hannah");
```

```
ourClassMates.get(2);
```

This code "fetches" index 2 which is "Hannah"

If we want to print Hannah,

```
System.out.print(ourClassmates.get(2));
```

Oct 28-7:52 PM

An Example of Each ...

```
ourClassMates.add("Mr. Henderson");
ourClassMates.add("Justin");
ourClassMates.add("Hannah");
```

```
ourClassMates.indexOf("Justin");
```

This code "fetches" index of "Justin" which is 2.

If we want to print the index location,

```
System.out.println(ourClassMates.indexOf("Justin"));
```

Oct 28-7:52 PM

An Example of Each ...

```
ourClassMates.add("Mr. Henderson");
ourClassMates.add("Justin");
ourClassMates.add("Hannah");
```

```
ourClassMates.size( );
```

This code "fetches" the length of the list which is 3.

If we want to print,

```
System.out.println(ourClassMates.size( ));
```

Oct 28-7:52 PM

An Example of Each ...

```
ourClassMates.add("Mr. Henderson");
ourClassMates.add("Justin");
ourClassMates.add("Hannah");
```

```
ourClassMates.set(0,"Dr. Hendermonger");
```

This code changes something in the list - updates!

If we want to print,

```
System.out.println(ourClassMates.get(0));
```

Oct 28-7:52 PM

Whooaaaaaa!!! There's more though!

```
ourClassMates.set(0,"Dr. Hendermonger");
```

This does change the first element to "Dr. Hendermonger"; however, it is a method that also returns the item that was replaced. So ...

```
ourClassMates.add(ourClassMates.set(0,"Dr. Hendermonger"));
```

This is a very nice way to add Dr. Hendermonger yet keep Mr. Henderson by putting him at the end of the list!

Oct 28-7:52 PM

Every Method Used Returns Something ...

Method	Returns
listName.add(x)	true
listName.remove(#)	Item that was removed
listName.get(#)	Item that it got
listName.indexOf(x)	Index integer number
listName.size( )	Integer size of list
listName.set(#,x)	Item that was replaced

Oct 28-7:52 PM

Drawing Comparisons ...

Array	ArrayList
Fixed size once made	Can be resized (real-time, auto)
Manually shift elements	Auto shift when insert/delete
Static, not changing	Ever-changing size?
ArrayIndexOutOfBoundsException	IndexOutOfBoundsException

Oct 28-7:52 PM

Things to be completing ...

- Unit 3 WS01-03 Should be done.
- Unit 3 WS04 Should be getting wrapped up.
- Unit 3 WS05 Assigned Today!

Oct 31-4:35 PM