

Java Data Structures: Arrays and ArrayLists

**CPSC 219: Introduction to Computer Science for Multidisciplinary
Studies II
Fall 2023**

Jonathan Hudson, Ph.D.
Instructor
Department of Computer Science
University of Calgary

Wednesday, 20 September 2023

Copyright © 2023



Where are lists?

- Java has no Python lists (or tuples)
- It has **Arrays**
 - share syntax with python lists like `index array[index] = 5;`
 - Size can't be changed
 - No syntax for slicing
 - No methods built in (can use `java.util.Arrays` library)

Arrays Library?

```
import java.util.Arrays;
```

- <https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/util/Arrays.html>
- Often useful for printing `Arrays.toString(array)`;
- For sorting `Arrays.sort(array)`;
- **No syntax for slicing**,
 - but can use `java.util.Arrays.copyOfRange(array, start, end)`;

Where are lists?

- Java has no Python lists (or tuples)
- It also has **ArrayLists**
 - Closest functionally to python lists
 - Syntax uses longer to type methods like `array_list.set(index, 5);`
 - Size can change
 - Everything via methods built in (like `.sort()`, `.size()`, etc.)
 - **No syntax slicing** but `array_list.subList(start, end)`

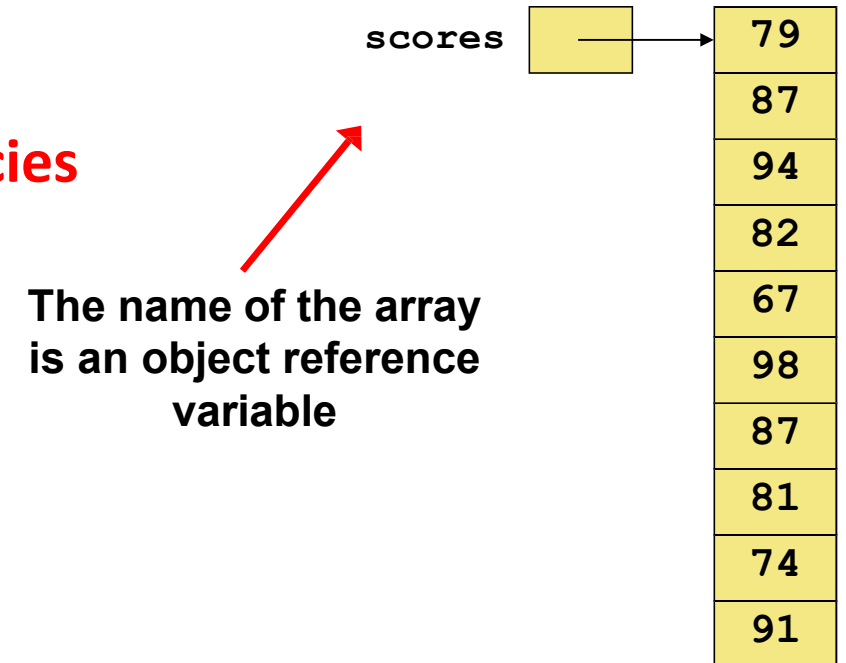
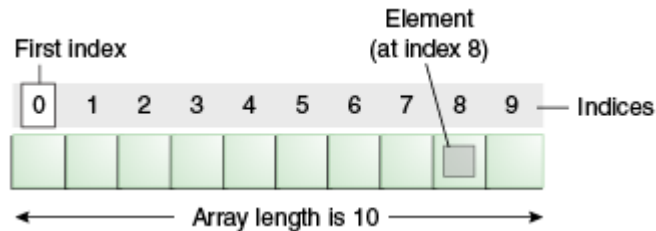
Where are lists?

- Java has no Python lists (or tuples)
- It has Arrays
 - share syntax with python lists like `index array[index] = 5;`
 - Size can't be changed
 - No methods built in (can use `java.util.Arrays` library)
 - No syntax slicing, but can use `java.util.Arrays.copyOfRange(array, start, end);`
- It has ArrayLists
 - Closest functionally to python lists
 - Syntax all through longer to type methods like `array_list.set(index, 5);`
 - Size can change
 - Everything via methods built in (like `.sort()`, `.size()`, etc.)
 - No syntax slicing but `array_list.subList(start, end)`

Arrays

Arrays

- Size is static (once you set it you are stuck with it)
 - Re-size by making new array and copying everything in
 - You can't remove primitives (just change their value)
 - You can remove objects but only by setting spot to null
- Are type limiting (all data must be the same)
- You can index into it (like python lists) , **no negative indicies**



The name of the array is an object reference variable

Arrays

```
int[] array;           //Create variable
array = new int[<size>; //Create array (filled with 0's)
int x = array[index]; //Index access
array[index] = <new value>; //Index assign

int[] array = {0,1,2,3,4,5,6,7,8,9}; //Create array all at once
```


String Arrays

```
String[] array;           //Create variable
array = new String[<size>; //Create array (filled with null)
String x = array[index];  //Index access
array[index] = <new value>; //Index assign

String[] array = {"0","1","2","3"}; //Create array all at once
```

2D Arrays

```
int[][] array; //Create variable
array = new int[<size1>][<size2>]; //Create 2D array (filled with 0's)
int x = array[index1][index2]; //Index access
array[index1][index2] = <new value>; //Index assign

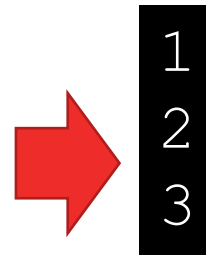
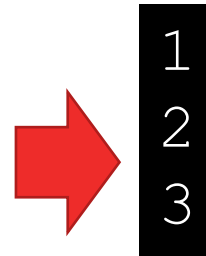
int[][] array = {{0,1,2},{3,4,5},{6,7,8}}; //Create array all at once
```

<size1> -> generally first dimension is called rows

<size2> -> generally second dimension is called columns

Looping Array

```
public static void main(String[] args) {  
    int[] array = {1,2,3};  
    for (int item : array) {  
        System.out.println(item);  
    }  
    for (int i = 0; i < array.length; i++) {  
        System.out.println(array[i]);  
    }  
}
```



Looping Array

```
public static void main(String[] args) {  
    int[][] array = {{1,2,3}, {4,5,6}};  
    for (int i = 0; i < array.length; i++) {  
        for (int j = 0; j < array[i].length; j++) {  
            System.out.println(array[i][j]);  
        }  
    }  
}
```



1
2
3
5
6
7

Looping Array

```
public static void main(String[] args) {  
    int[][] array = {{1,2,3}, {4,5,6}};  
    for (int row = 0; row < array.length; row++) {  
        for (int col = 0; col < array[row].length; col++) {  
            System.out.println(array[row][col]);  
        }  
    }  
}
```



1
2
3
5
6
7

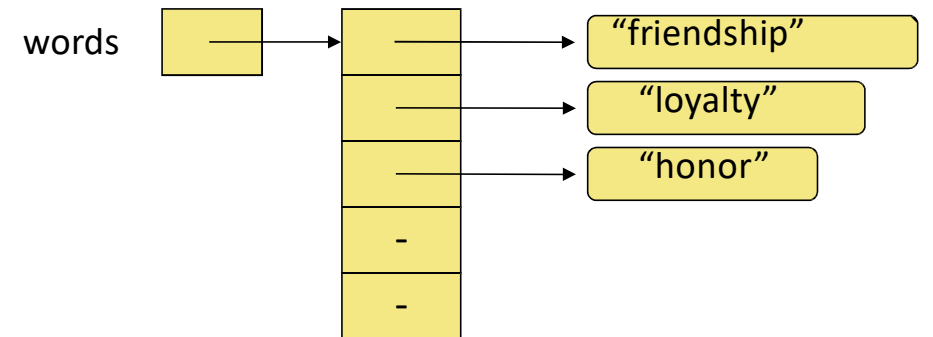
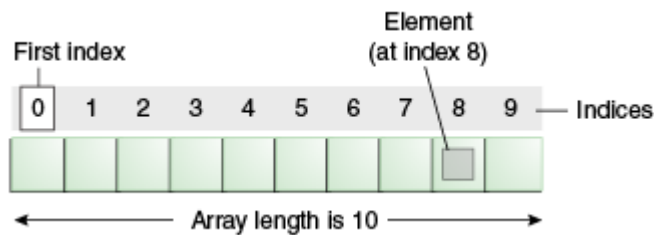
ArrayLists

ArrayLists

- Size changes as needed (technically an ArrayList is a wrapper of a hidden array)
- Only stores object types
- List size changes as you add/remove things
- Can be type limiting, using <Type> generics, otherwise everything is Object
- import java.util.ArrayList;
- <https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/util/ArrayList.html>

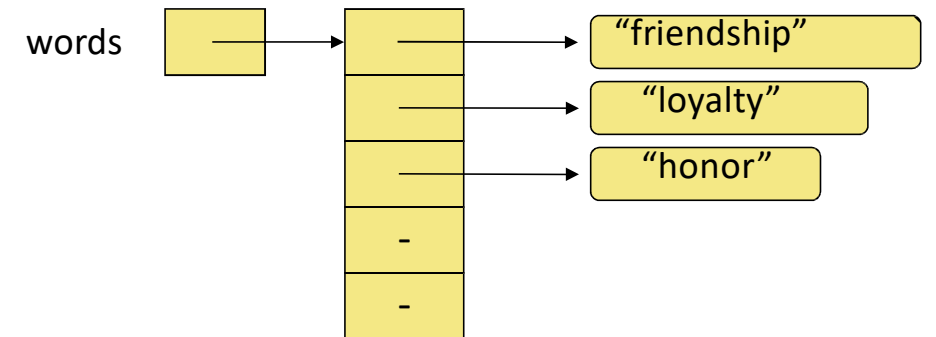
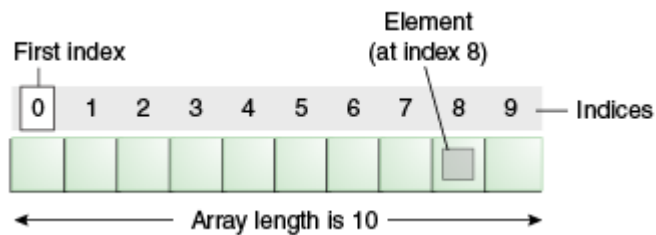
ArrayLists

```
ArrayList<String> list; //Create variable  
list = new ArrayList<String>(); //Create array (size=0)  
String x = list.get(index); //Index access  
list.set(index, <new value>); //Index assign
```



ArrayLists (Non-generic)

```
ArrayList list; //Create variable  
list = new ArrayList(); //Create array (size=0)  
Object x = list.get(index); //Index access  
String y = (String) x;  
list.set(index, <new value>); //Index assign
```



ArrayLists - methods

Java	Java (desc)	Python	Python (desc)
<code>list.add(index, item);</code>	Insert item at index, shift left	<code>list.insert(index, item)</code>	same
<code>list.addAll(list);</code>	Insert all at index	<code>list.extend(list)</code>	same
<code>list.clear();</code>	Clear all	<code>list.clear()</code>	same
<code>list.contains(item);</code>	Check if item in list	<code>item in list</code>	same
<code>list.indexOf(item);</code>	Get first index of item	<code>list.index(item)</code>	same
<code>list.remove(index);</code>	Remove by index	<code>list.pop(index)</code>	same
<code>list.remove(item);</code>	Remove by item	<code>list.remove(item)</code>	same
<code>list.size();</code>	Size of list	<code>len(list)</code>	same
<code>Collections.sort(list);</code>	Sort list*	<code>list.sort()</code>	same

*import java.util.Collections;

Simple operations

```
ArrayList<String> list = new ArrayList<String>();  
list.add("3"); list.add("2"); list.add("1");
```

```
System.out.println(list);
```

[3, 2, 1]

```
Collections.sort(list);
```

[1, 2, 3]

```
System.out.println(list);
```

1

```
System.out.println(list.indexOf("2"));
```

```
list.remove("2");
```

[1, 3]

```
System.out.println(list);
```

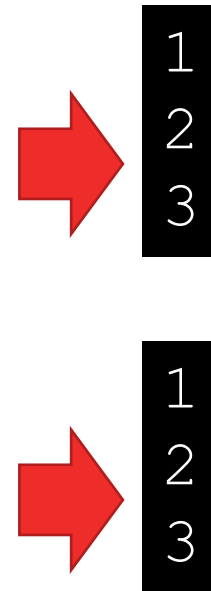
```
list.remove(0);
```

[3]

```
System.out.println(list);
```

Looping ArrayList

```
ArrayList<String> list = new ArrayList<String>();  
list.add("1"); list.add("2"); list.add("3");  
for (String item : list) {  
    System.out.println(item);  
}  
for (int i = 0; i < list.size(); i++) {  
    System.out.println(list.get(i));  
}
```



java.util.Collections

- Like java.util.Arrays (but for things like ArrayList, and HashSet/HashMap)
- <https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/util/Collections.html>
- min()
- max()
- sort()
- shuffle()
- reverse()
- replaceAll()

Onward to ... HashSets and HashMaps.

Jonathan Hudson
jwhudson@ucalgary.ca
<https://pages.cpsc.ucalgary.ca/~jwhudson/>



UNIVERSITY OF
CALGARY