

CMPSC 100

Computational Expression

Arrays

- Provide similar basic structures as ArrayLists
 - However, they are *very* basic--we do not have:
 - indexOf
 - set
 - get
 - add
 - ...
- If there's no useful functionality, why do they exist!?
 - It's a question of *purpose*

Array vs. ArrayList

ArrayList

Pros

- Good for lists of unknown size
- Has several built-in “services” (get, add, indexOf)

Cons

- Can’t use primitives (conversions using wrapper classes “cost”)
- Many libraries don’t take ArrayList parameters

Array

Pros

- Great for lists of known size
- Can be “faster” or less “costly” for memory
- Appear often “in the wild”
- Have simpler “dimensionality”

Cons

- No built-in “services”
- Syntax can look somewhat cryptic

Syntax

```
int[] = new int[10];
```

```
char[] = new char[4];
```

```
String[] sentences = new String[145];
```

```
int[][] = new int[8][8];
```

```
int[9] numbers = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

```
String[3] articles = {"a", "an", "the"};
```

Syntax

```
int[9] numbers = {1, 3, 2, 4, 5, 7, 6, 8, 9};  
String[3] articles = {"a", "an", "the"};
```

```
numbers[0]
```

```
> 1
```

```
articles[1];
```

```
> "an"
```

```
articles[3]
```

```
> ArrayIndexOutOfBoundsException
```

“Dimensionality”

```
// Tic-tac-toe board  
int[][] board = int[3][3]
```

```
// Chess board  
int[][] board = int[8][8]
```

```
// ArrayList tic-tac-toe board  
ArrayList<ArrayList<Integer>> board = new ArrayList<Integer>();
```

Exercise

Perform a `git pull download master`

Navigate to the `class-activities/6-november` folder.

Exercise

We'll revisit our M&Ms count file and do the following:

- Read the file to a `String[][]` called “table”
- Calculate the averages for each column
- Display the file using that two-dimensional array
- Display the averages below the table


```
String[][] table = new String[117][8];  
int row = 0;  
Scanner data;
```

```
while(input.hasNext()){  
    int column = 0;  
    String line = input.nextLine();  
    data = new Scanner(line);  
    data.useDelimiter(",");  
    while(data.hasNext()){  
        String field = data.next();  
        table[row][column] = field;  
        column++;  
    }  
    row++;  
}
```

```
int[] averages = new int[8];
for(int r = 0; r < table.length; r++) {
    for(int c = 0; c < table[r].length; c++){
        System.out.print(table[r][c] + "\t");
        if (r > 0 & c > 0) {
            averages[c] += Integer.parseInt(table[r][c]);
        }
    }
    System.out.println();
}
```

```
System.out.print("Aves. \t");  
for(int c = 1; c < averages.length; c++){  
    System.out.print(averages[c]/(row-1) + "\t");  
}
```

Test using `gradle run`