

LEC 15

CSE 121

2D Arrays

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

Talk to your neighbors:

*What computer application (website, app)
do you use the most on a daily basis?*

Music:  [CSE 121 25au Lecture Tunes](#) 

Instructors: Brett Wortzman & James Weichert

TAs:	Trey	Ava	Caleb	Elden	Anya
	Amogh	Reese	Anum	Suyash	Minh
	Samrutha	Hayden	Abdul	Sthiti	TJ
	Dalton	Aki	Janvi	Paul	Zach
	Ailsa	Spencer	Navya	Shayna	Cayden
	Ryan	Savannah	Sam	Jesse	Johnathan
	Anant	Tamsyn	Jessica	Nhan	

Announcements, Reminders

- C3 released, due **Tuesday, November 18th**
- R5 due next **Thursday, November 20th** (eligible: **P1**, C2, P2)
 - P1 cycling out of eligibility after R5
- Quiz 2 on **Thursday, November 20th**
 - can't make it? email instructors *before* your quiz!
- In the future: Final Exam on **December 10th from 12:30-2:20 PM**

P2 Reflections: Algorithms and Bias

- We're "automating human biases into rules"
- We introduce clear socioeconomic bias by taking into account factors that are not relevant to healthcare
- **"The idea of prioritizing patients is inherently biased"**
- "Our algorithmic assumes things, like that people under a certain age and over a certain age are inherently more at risk"
- We're prioritizing (optimizing for) "speed over fairness"
- Priority levels are inherently limited — "serious symptoms could easily fly under the radar"

P2 Reflections: Addressing Bias

- Have *more* “objective medical vitals”
- Have *less* input
- Remove hardcoded thresholds
- “We could try adding AI to decide on the urgency of the patient”
- “We should add a ‘fairness adjustment’ for medically underserved communities to correct for historical healthcare disparities”
- We should “check fairness by **testing** ‘what if’ cases”
- **“We could add a human review step”**

Can we eliminate *all* bias from this algorithm?



Practice: Think

sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```

Four lines of output are produced by this code. What would those four lines be?



Practice: Pair

sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```

Four lines of output are produced by this code. What would those four lines be?



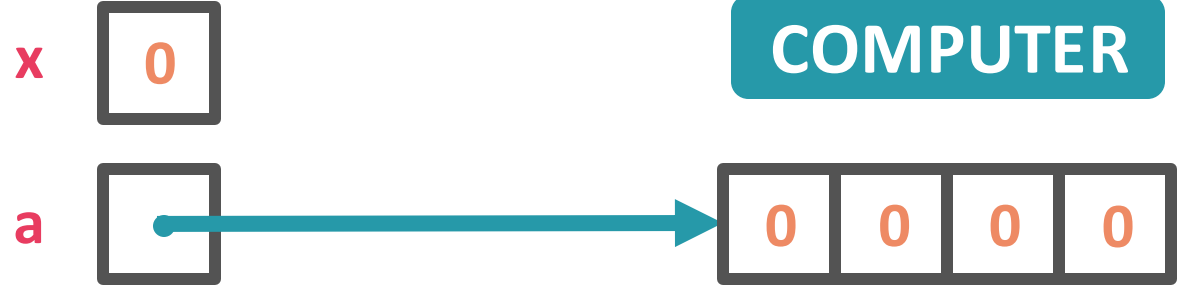
Practice: Pair



sli.do

#cse121

```
public static void main(String[] args) {  
    ➔ int x = 0;  
    ➔ int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```





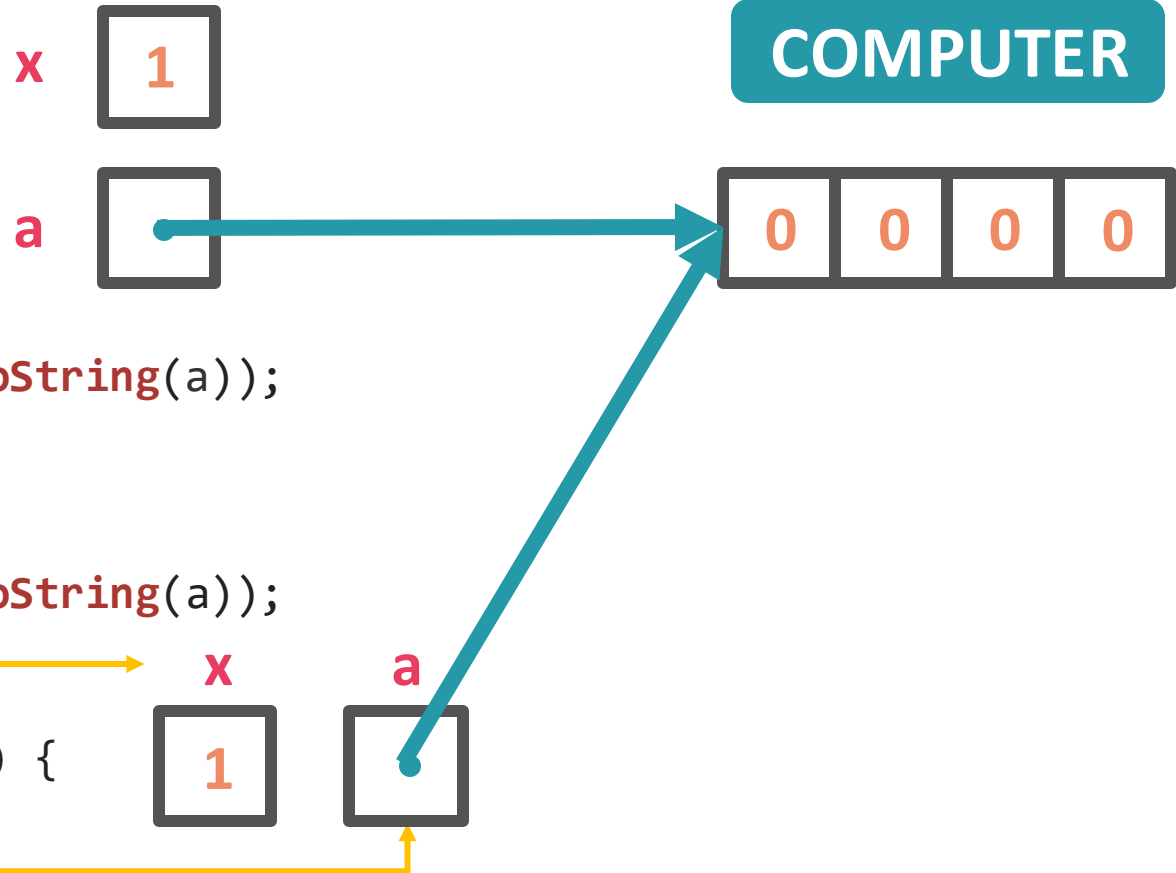
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    ➔ x++;  
  
    ➔ mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    ➔ x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```





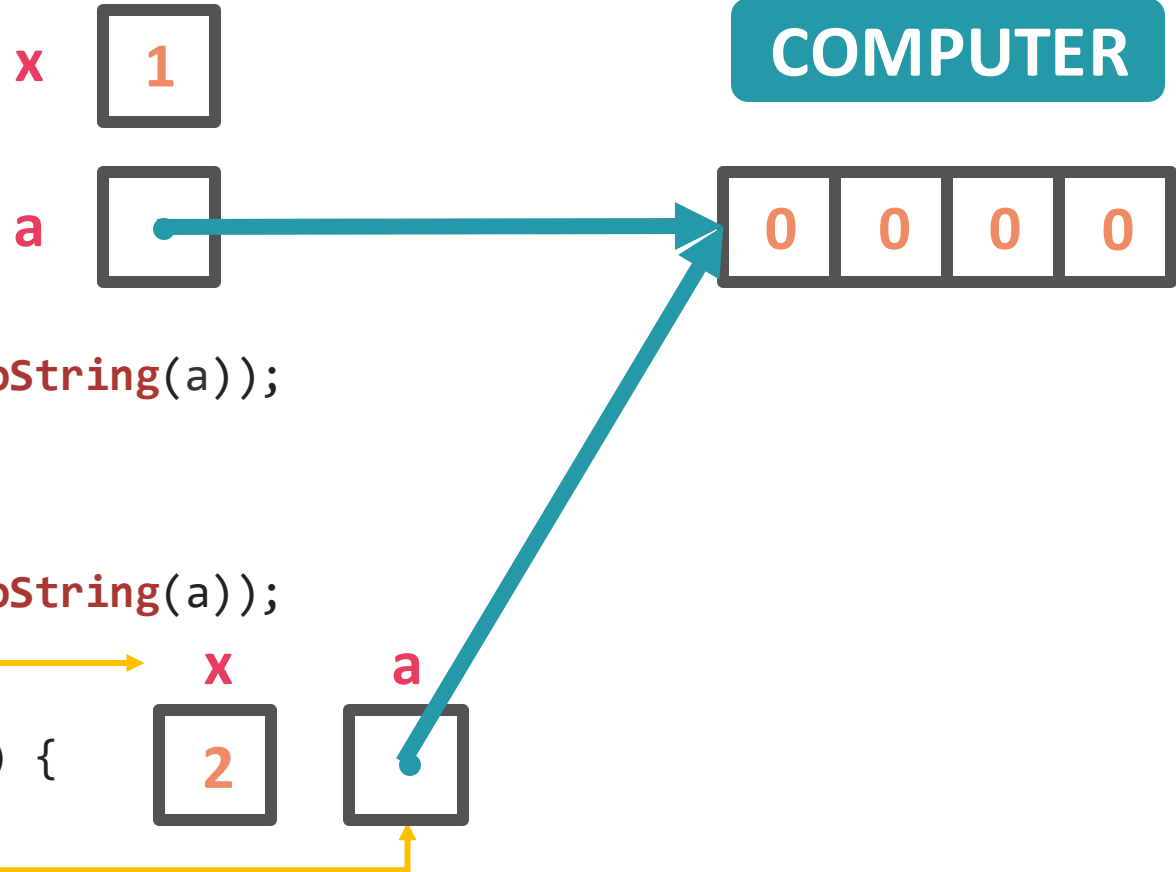
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```





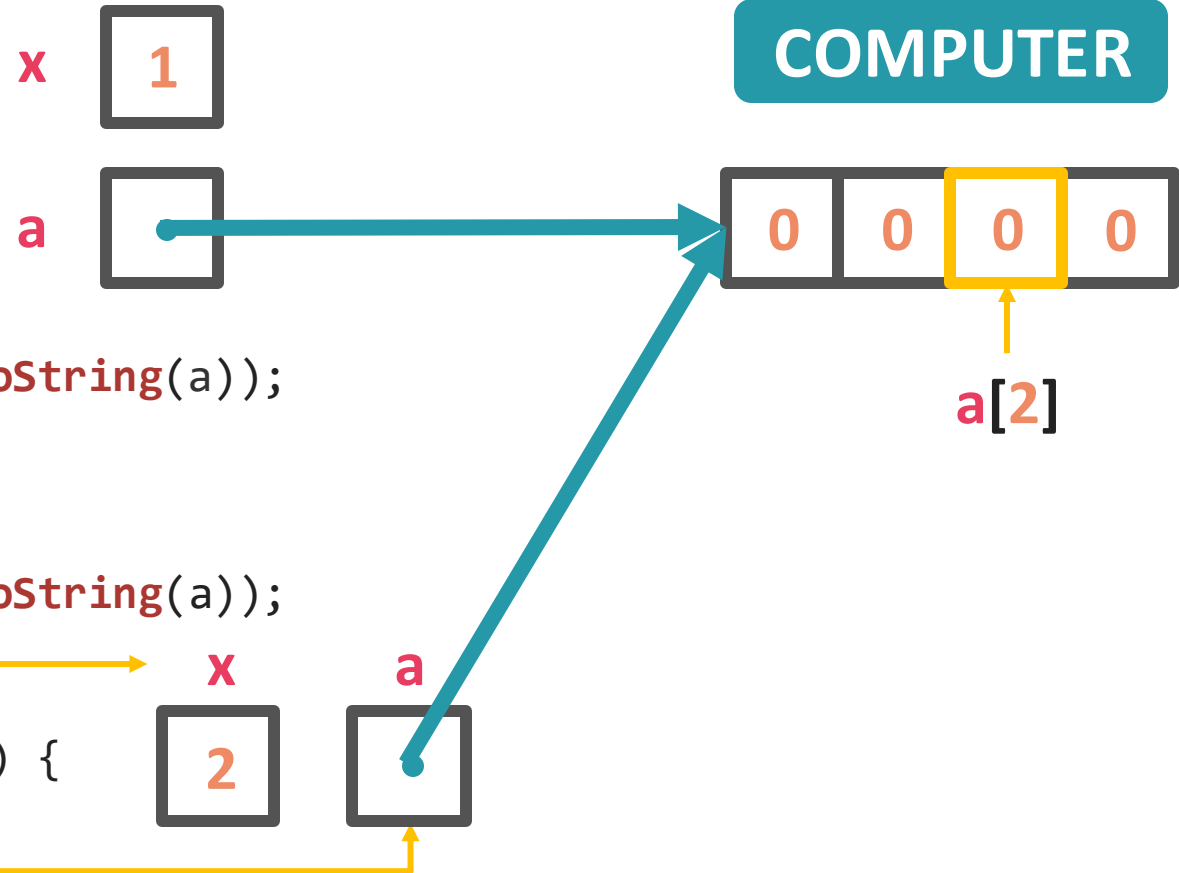
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```





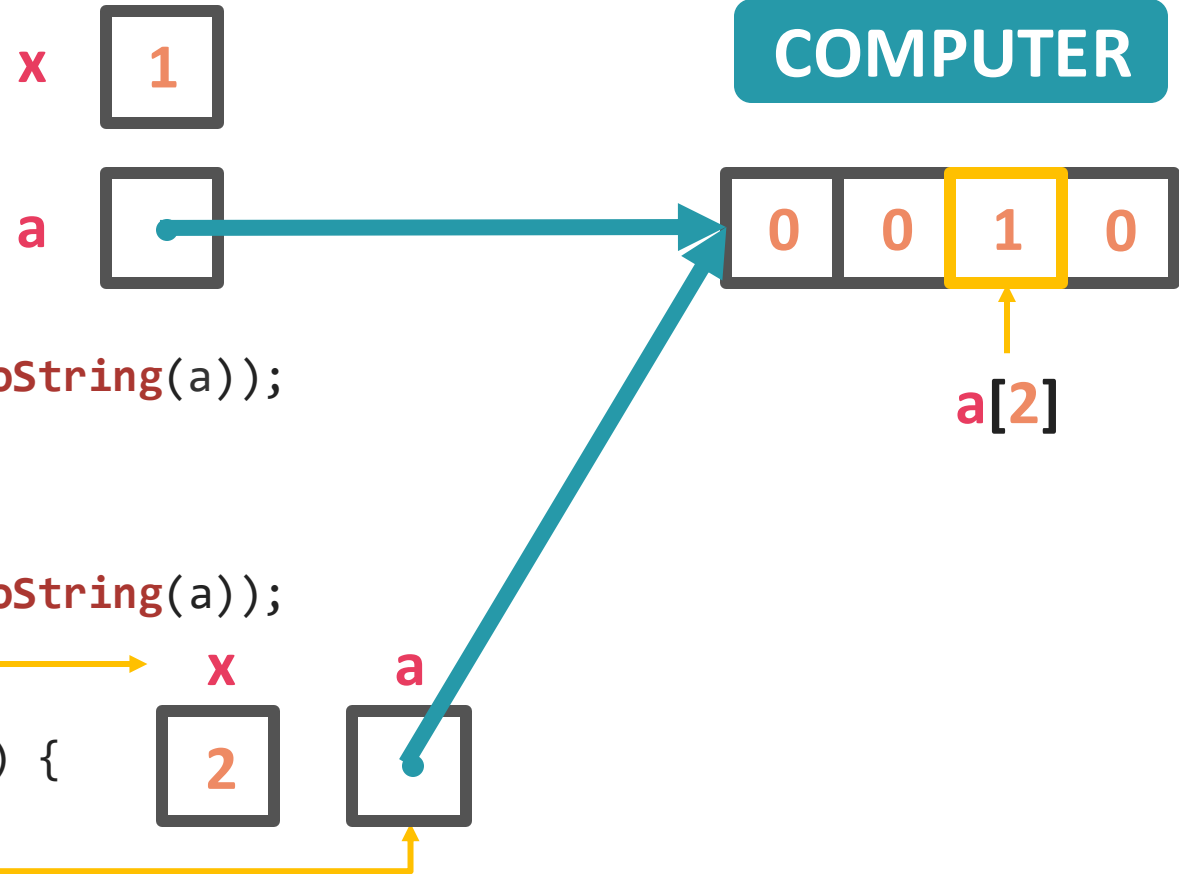
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```





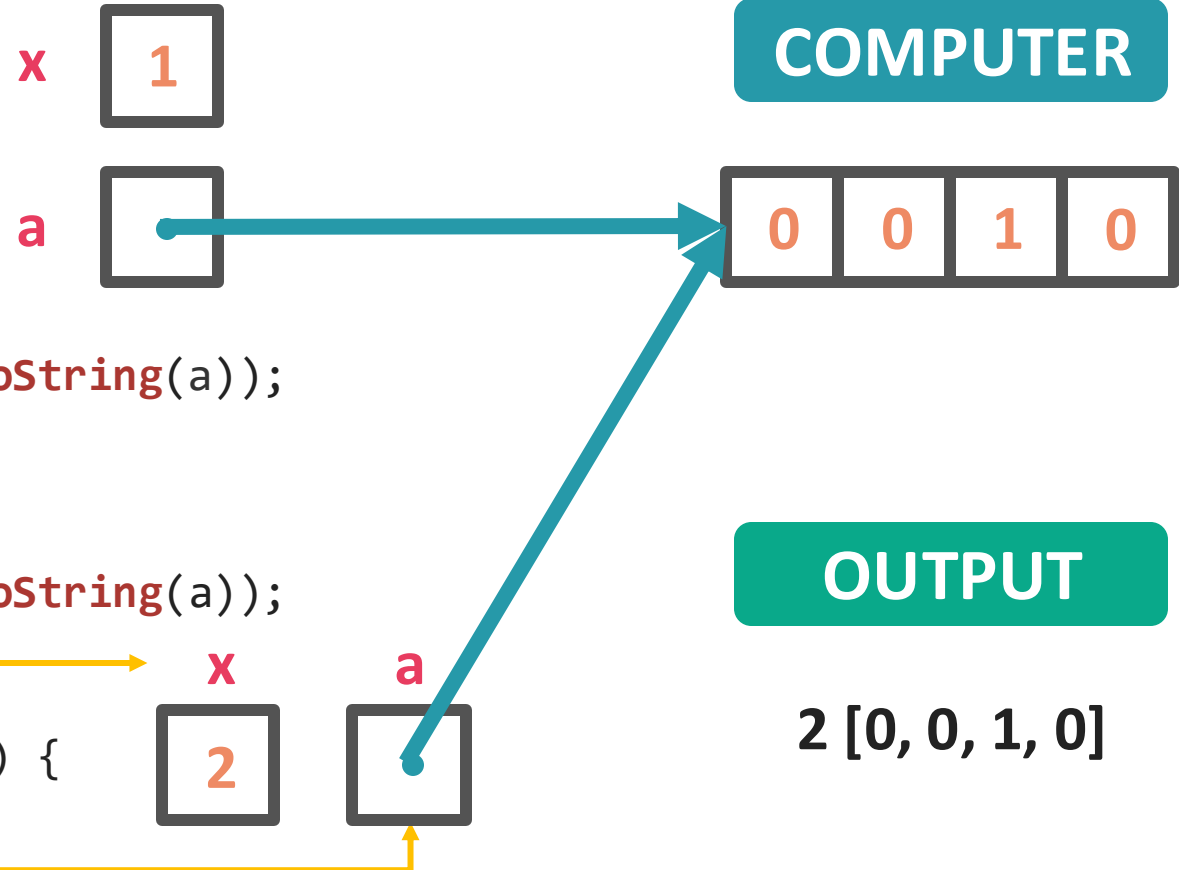
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```





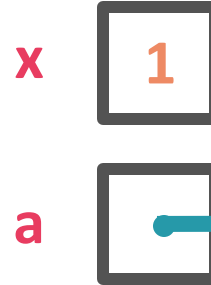
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;
```



COMPUTER



```
    mystery(x, a);
```

```
    System.out.println(x + " " + Arrays.toString(a));
```

```
    x++;
```

```
    mystery(x, a);
```

```
    System.out.println(x + " " + Arrays.toString(a));
```

```
}
```

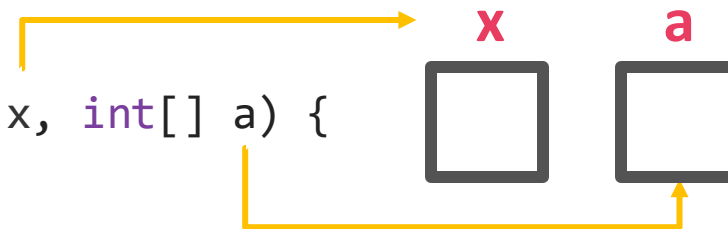
```
public static void mystery(int x, int[] a) {
```

```
    x++;
```

```
    a[x]++;
```

```
    System.out.println(x + " " + Arrays.toString(a));
```

```
}
```



OUTPUT

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]



Practice: Think



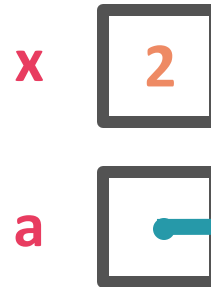
sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}
```

➡ x++;
➡ **mystery**(x, a);
System.out.println(x + " " + Arrays.toString(a));
}

```
public static void mystery(int x, int[] a) {  
    ➡ x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```

**COMPUTER****OUTPUT**

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]



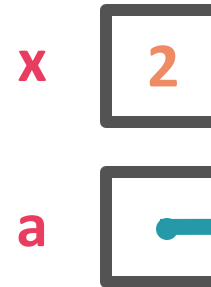
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```



COMPUTER



OUTPUT

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]



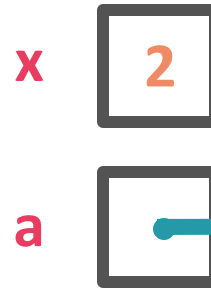
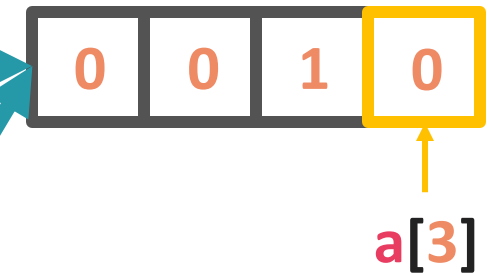
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```

**COMPUTER****OUTPUT**

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]



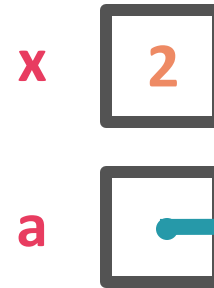
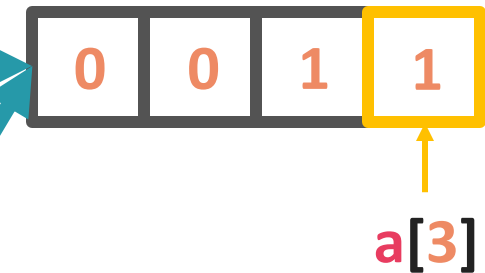
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```

**COMPUTER****OUTPUT**

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]



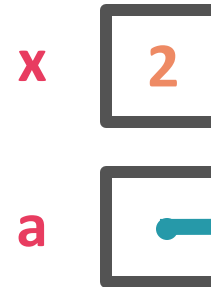
Practice: Think



sli.do

#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}  
  
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```



COMPUTER



OUTPUT

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]

3 [0, 0, 1, 1]



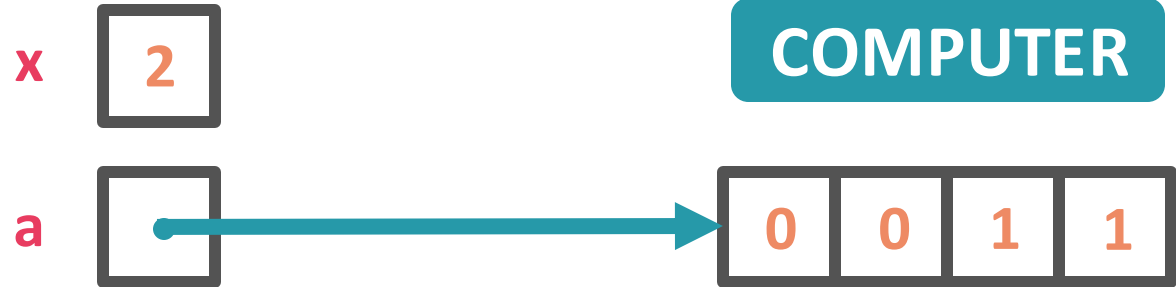
Practice: Think



sli.do

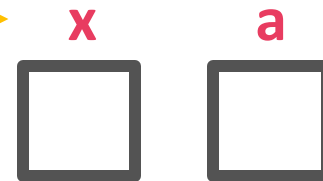
#cse121

```
public static void main(String[] args) {  
    int x = 0;  
    int[] a = new int[4];  
    x++;  
  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
  
    x++;  
    mystery(x, a);  
    System.out.println(x + " " + Arrays.toString(a));  
}
```



COMPUTER

```
public static void mystery(int x, int[] a) {  
    x++;  
    a[x]++;  
    System.out.println(x + " " + Arrays.toString(a));  
}
```



OUTPUT

2 [0, 0, 1, 0]

1 [0, 0, 1, 0]

3 [0, 0, 1, 1]

2 [0, 0, 1, 1]

PCM Review: 2D Arrays

```
int[][] a = new int[4][3];
```

type name array creation code

An array of arrays!

The type of each individual element is another array!

- Your first example of “**nested** data structures”
- There will be more if you take CSE 122!

int[][]

double[][]

String[][]

boolean[][]

char[][]

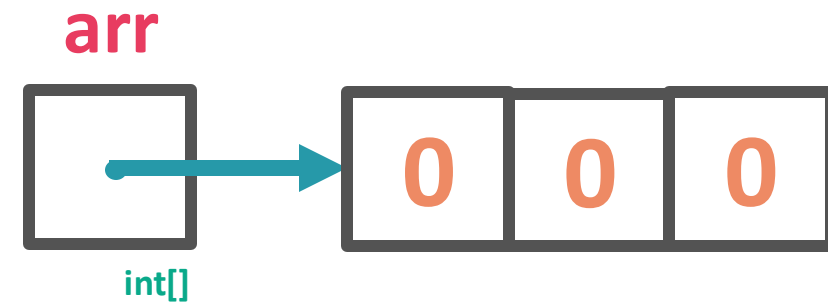
Recall: Arrays

type name value

```
int[] arr = new int[3];
```

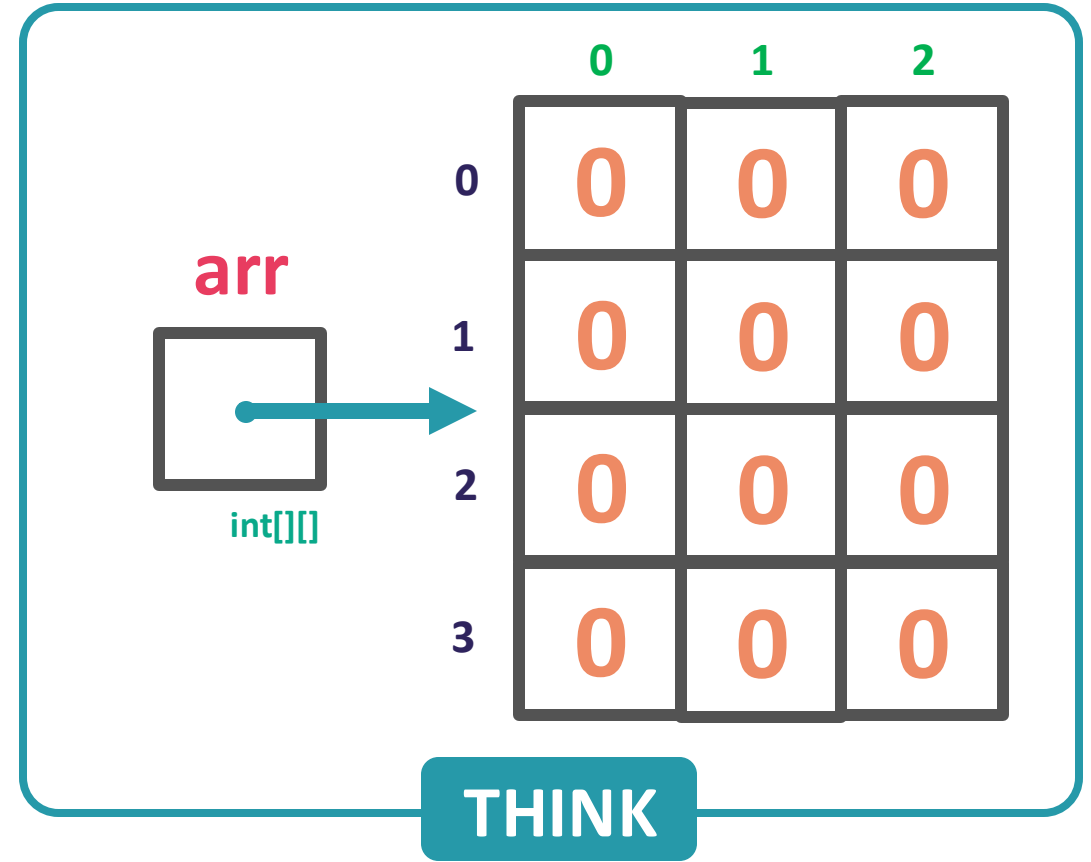
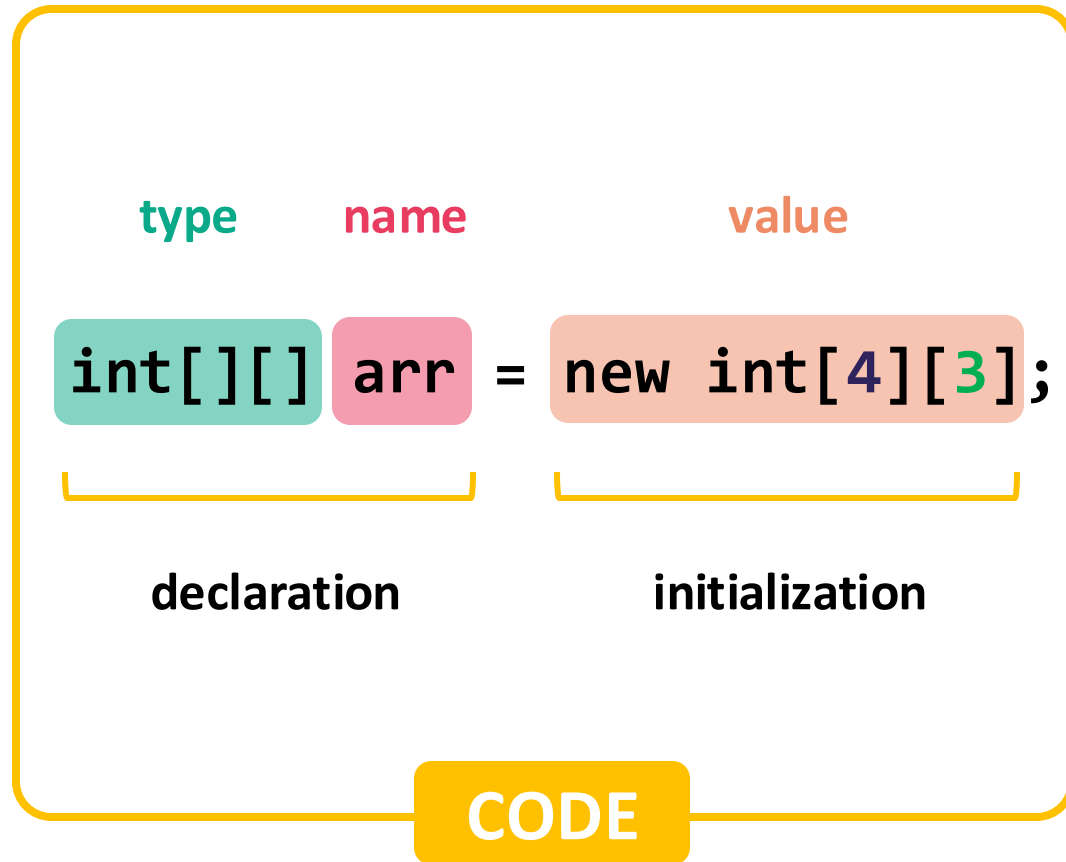
declaration initialization

CODE



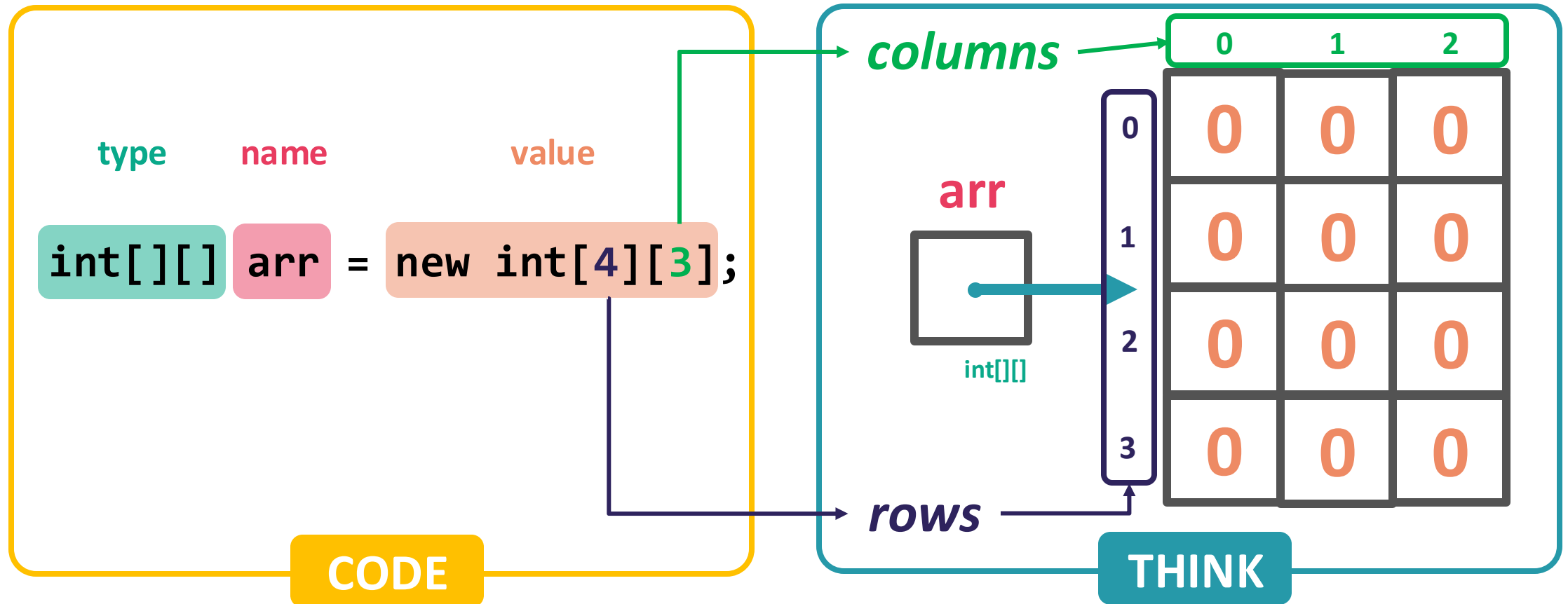
THINK

PCM Review: 2D Arrays as Rows and Columns



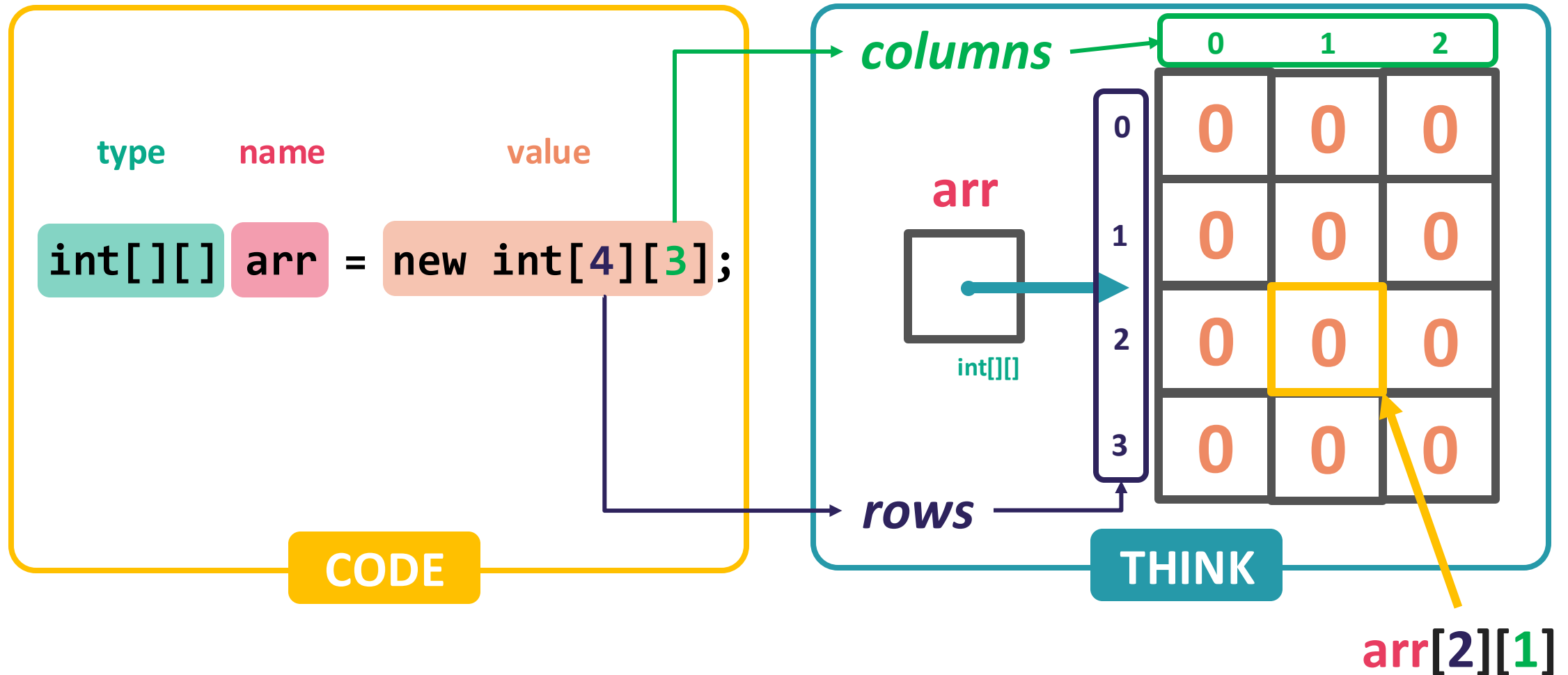
PCM Review: 2D Arrays as Rows and Columns

Individual elements are **row-first**, then **column** (“row-major”)

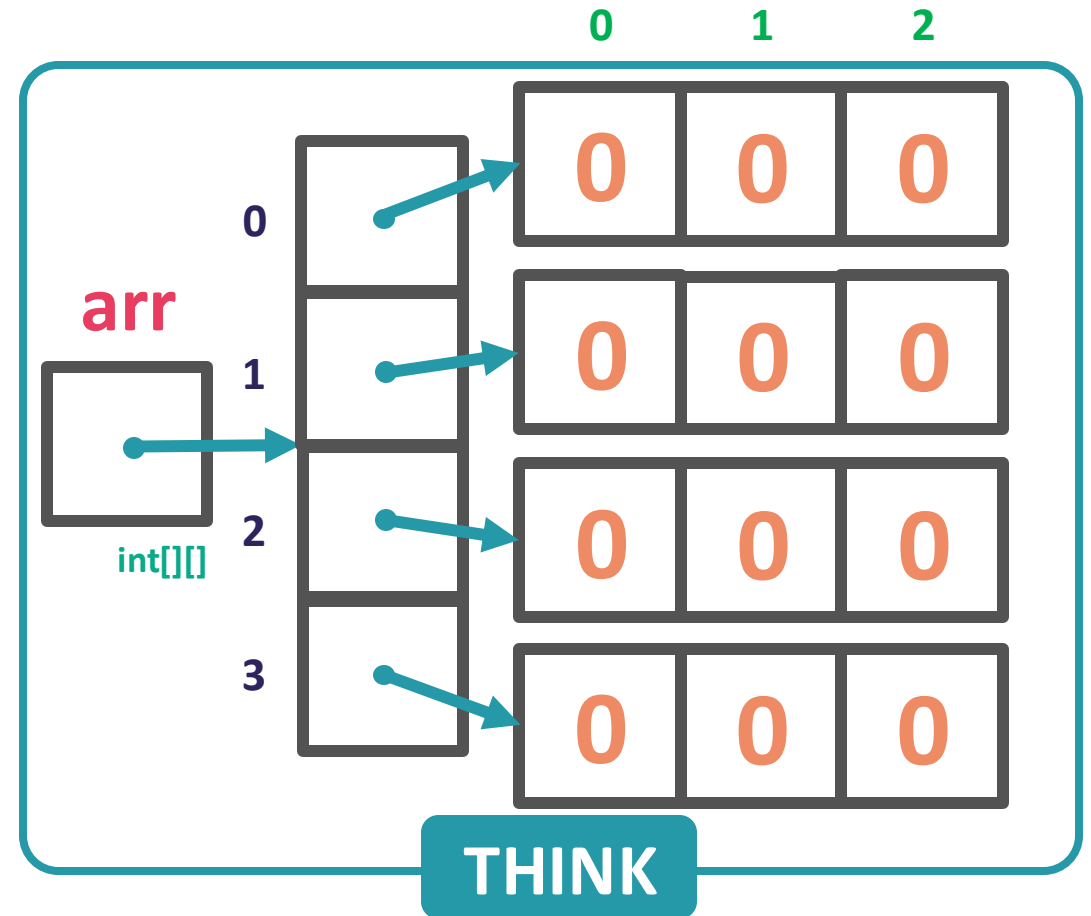
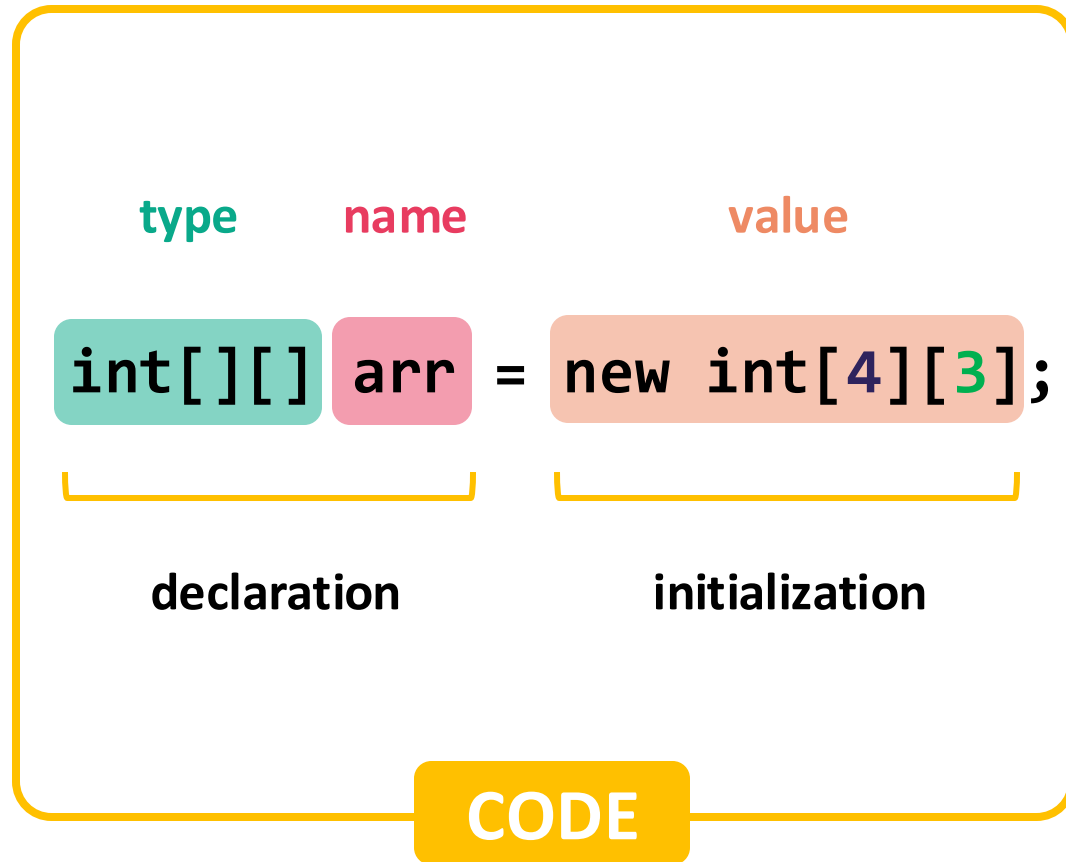


PCM Review: 2D Arrays as Rows and Columns

Individual elements are **row-first**, then **column** (“row-major”)



A More Accurate View (Reference Semantics)



PCM Review: 2D Array Traversals

for each row...

```
for (int i = 0; i < list.length; i++) {  
    for (int j = 0; j < list[i].length; j++) {  
        // do something with list[i][j]  
    }  
}
```

for each element within a row...

Arrays Utility Class (and methods)

Method	Description
<code>Arrays.toString(array);</code>	Returns a <code>String</code> representing the array, such as <code>"[10, 30, -25, 17]"</code>
<code>Arrays.equals(array1, array2);</code>	Returns <code>true</code> if the two arrays contain the same elements in the same order
<code>Arrays.deepToString(array);</code>	Returns a <code>String</code> representing the array; if the array contains other arrays as elements, the <code>String</code> represents their contents, and so on. For example, <code>"[[99, 151], [30, 5]]"</code>
<code>Arrays.deepEquals(array1, array2);</code>	Returns <code>true</code> if the two arrays contain the same elements in the same order; if the array(s) contain other arrays as elements, their contents are tested for equality, and so on.

Applications of 2D Arrays

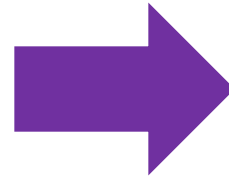
- **Matrices**
 - Useful in *so many* applications in math, engineering, and statistics
 - Fundamental to machine learning & AI
 - P3 is one real-life application of this! (bioinformatics)
- Board games
 - e.g. chess, checkers, tic-tac-toe, sudoku
- Tabular or grid-like data
 - e.g. scorekeeping, gradebook, census data
- Image processing

matrixAdd

23	96	18	4	64
45	40	18	44	34
92	13	77	71	12



70	73	66	79	39
91	75	73	99	47
27	64	21	34	1



matrixAdd: first row

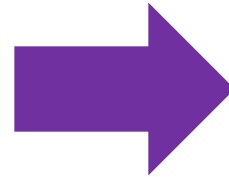
23	96	18	4	64
45	40	18	44	34
92	13	77	71	12



70	73	66	79	39
91	75	73	99	47
27	64	21	34	1

i: 0

j: 0



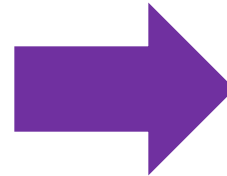
matrixAdd: 0,0

<u>23</u>	96	18	4	64
45	40	18	44	34
92	13	77	71	12



<u>70</u>	73	66	79	39
91	75	73	99	47
27	64	21	34	1

i: 0
j: 0



<u>93</u>				

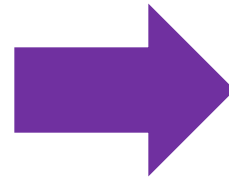
matrixAdd: 0,1

23	<u>96</u>	18	4	64
45	40	18	44	34
92	13	77	71	12



70	<u>73</u>	66	79	39
91	75	73	99	47
27	64	21	34	1

i: 0
j: 1



93	<u>169</u>			

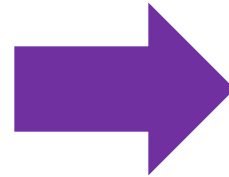
matrixAdd: 0,2

23	96	<u>18</u>	4	64
45	40	18	44	34
92	13	77	71	12



<u>70</u>	73	<u>66</u>	79	39
91	75	73	99	47
27	64	21	34	1

i: 0
j: 2



93	169	<u>84</u>		

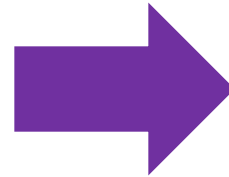
matrixAdd: 0,3

23	96	18	<u>4</u>	64
45	40	18	44	34
92	13	77	71	12



70	73	66	<u>79</u>	39
91	75	73	99	47
27	64	21	34	1

i: 0
j: 3



93	169	84	<u>83</u>	

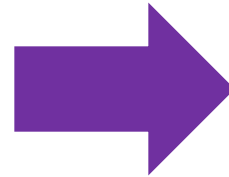
matrixAdd: 0,4

23	96	18	4	<u>64</u>
45	40	18	44	34
92	13	77	71	12



70	73	66	79	<u>39</u>
91	75	73	99	47
27	64	21	34	1

i: 0
j: 4



93	169	84	83	<u>103</u>

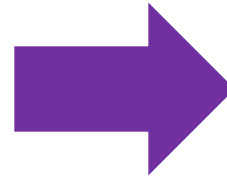
matrixAdd: 1,0

23	96	18	4	64
<u>45</u>	40	18	44	34
92	13	77	71	12



70	73	66	79	39
<u>91</u>	75	73	99	47
27	64	21	34	1

i: 1
j: 0



93	169	84	83	103
<u>136</u>				

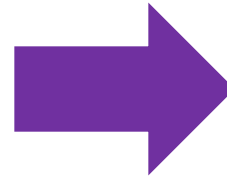
matrixAdd: 1,1

23	96	18	4	64
45	<u>40</u>	18	44	34
92	13	77	71	12



70	73	66	79	39
91	<u>75</u>	73	99	47
27	64	21	34	1

i: 1
j: 1



93	169	84	83	103
136	<u>115</u>			

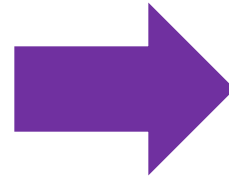
matrixAdd: 1,2

23	96	18	4	64
45	40	<u>18</u>	44	34
92	13	77	71	12



70	73	66	79	39
91	75	<u>73</u>	99	47
27	64	21	34	1

i: 1
j: 2



93	169	84	83	103
136	115	<u>91</u>		

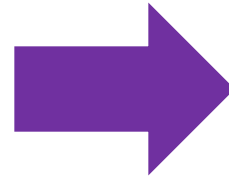
matrixAdd: 1,3

23	96	18	4	64
45	40	18	<u>44</u>	34
92	13	77	71	12



70	73	66	79	39
91	75	73	<u>99</u>	47
27	64	21	34	1

i: 1
j: 3



93	169	84	83	103
136	115	91	<u>143</u>	

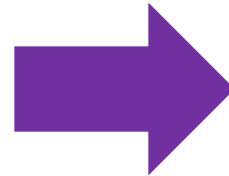
matrixAdd: 1,4

23	96	18	4	64
45	40	18	44	<u>34</u>
92	13	77	71	12



70	73	66	79	39
91	75	73	99	<u>47</u>
27	64	21	34	1

i: 1
j: 4



93	169	84	83	103
136	115	91	143	<u>81</u>

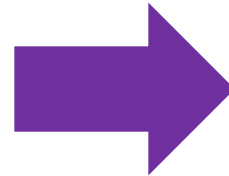
matrixAdd: 2,0

23	96	18	4	64
45	40	18	44	34
<u>92</u>	13	77	71	12



70	73	66	79	39
91	75	73	99	47
<u>27</u>	64	21	34	1

i: 2
j: 0



93	169	84	83	103
136	115	91	143	81
<u>119</u>				

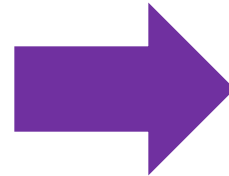
matrixAdd: 2,1

23	96	18	4	64
45	40	18	44	34
92	<u>13</u>	77	71	12



70	73	66	79	39
91	75	73	99	47
27	<u>64</u>	21	34	1

i: 2
j: 1



93	169	84	83	103
136	115	91	143	81
119	<u>77</u>			

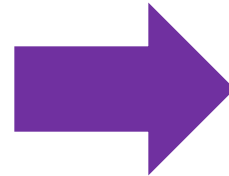
matrixAdd: 2,2

23	96	18	4	64
45	40	18	44	34
92	13	<u>77</u>	71	12



70	73	66	79	39
91	75	73	99	47
27	64	<u>21</u>	34	1

i: 2
j: 2



93	169	84	83	103
136	115	91	143	81
119	77	<u>98</u>		

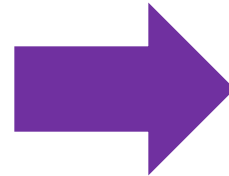
matrixAdd: 2,3

23	96	18	4	64
45	40	18	44	34
92	13	77	<u>71</u>	12



70	73	66	79	39
91	75	73	99	47
27	64	21	<u>34</u>	1

i: 2
j: 3



93	169	84	83	103
136	115	91	143	81
119	77	98	<u>105</u>	

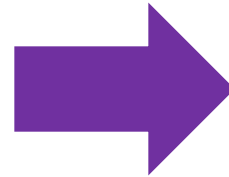
matrixAdd: 2,4

23	96	18	4	64
45	40	18	44	34
92	13	77	71	<u>12</u>



70	73	66	79	39
91	75	73	99	47
27	64	21	34	<u>1</u>

i: 2
j: 4



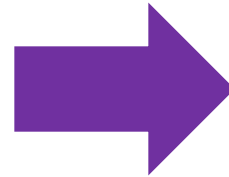
93	169	84	83	103
136	115	91	143	81
119	77	98	105	<u>13</u>

matrixAdd: finished!

23	96	18	4	64
45	40	18	44	34
92	13	77	71	12



70	73	66	79	39
91	75	73	99	47
27	64	21	34	1



93	169	84	83	103
136	115	91	143	81
119	77	98	105	13

readData: processing input

```
How many days' data would you like to input? 3
```

```
Next day's data:
```

```
    Temperature in Seattle? 44
```

```
    Temperature in Tacoma? 40
```

```
    Temperature in Bothell? 43
```

```
Next day's data:
```

```
    Temperature in Seattle? 42
```

```
    Temperature in Tacoma? 40
```

```
    Temperature in Bothell? 44
```

```
Next day's data:
```

```
    Temperature in Seattle? 42
```

```
    Temperature in Tacoma? 41
```

```
    Temperature in Bothell? 43
```

```
...
```

readData: days



How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1			
2			
3			

readData: 1, Seattle

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44		
2			
3			

readData: 1, Tacoma

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	
2			
3			

readData: 1, Bothell

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	43
2			
3			

readData: 2, Seattle

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma?

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	43
2	42		
3			

readData: 2, Tacoma

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 40

Temperature in Bothell?

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	43
2	42	40	
3			

readData: 2, Bothell

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 40

Temperature in Bothell? 44

Next day's data:

Temperature in Seattle?

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	43
2	42	40	44
3			

readData: 2, Seattle

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 40

Temperature in Bothell? 44

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma?

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	43
2	42	40	44
3	42		



readData: 2, Tacoma

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 40

Temperature in Bothell? 44

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 41

Temperature in Bothell?

...

	Seattle	Tacoma	Bothell
1	44	40	43
2	42	40	44
3	42	41	



readData: 2, Bothell

How many days' data would you like to input? 3

Next day's data:

Temperature in Seattle? 44

Temperature in Tacoma? 40

Temperature in Bothell? 43

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 40

Temperature in Bothell? 44

Next day's data:

Temperature in Seattle? 42

Temperature in Tacoma? 41

Temperature in Bothell? 43

...

	Seattle	Tacoma	Bothell
1	44	40	43
2	42	40	44
3	42	41	43



computeAverages

How many days' data would you like to input? 3

...

The average values for each location were
[42.666666666666664, 40.333333333333336,
43.333333333333336]

	Seattle	Tacoma	Bothell
1	44	40	43
2	42	40	44
3	42	41	43



42.667	40.333	43.333
--------	--------	--------



Average of Seattle
temperatures
(44 + 42 + 42) / 3