

LEC 13

CSE 121

Arrays

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

Talk to your neighbors:

*What is your favorite potato-based dish?
(e.g. fries, aloo gobi, gnocchi)*

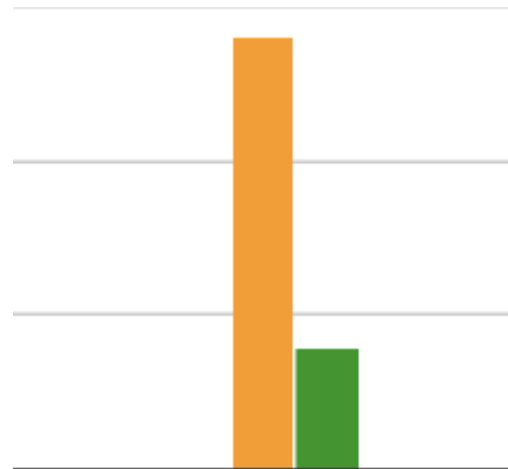
Instructors: Hayden Feeney**TAs:** Anisha Savannah William
Cayden Tamsyn

Announcements, Reminders

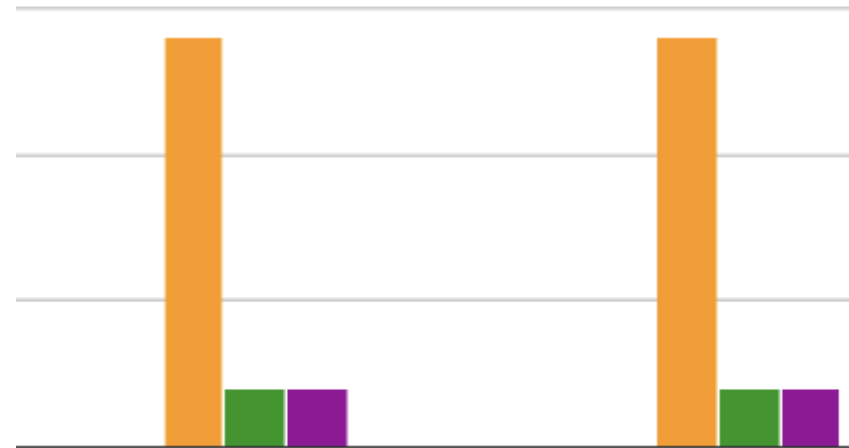
- P2 due tomorrow, **Thursday August 6th**
- R3 due tomorrow, **Thursday, August 6th** (eligible: **P0**, C1, P1, C2)
- Quiz 2 next week, **Thursday, August 13th**
 - Includes everything through today's lecture
 - Can't make it? Email Hayden *before* your quiz!
- Quiz 1 grades released before Quiz 2
- Final Exam is in two weeks, **Friday, August 21st**

Lecture feedback (1/2)

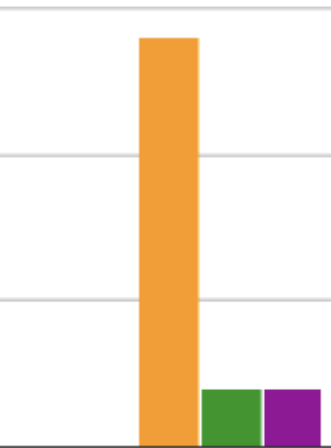
Very slow A bit slow Just right A little fast Way too fast



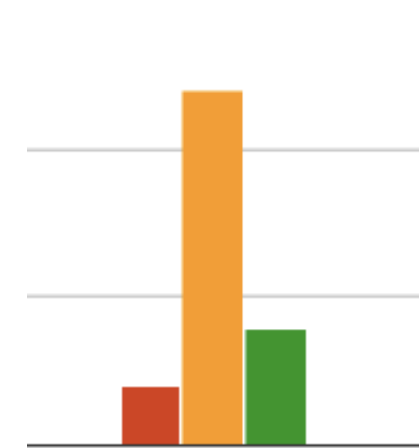
Review of PCM concepts



Coding examples
(worktime)



Coding examples
(walkthrough)



Think pair share (think
time)

Lecture feedback (2/2)

- Think Pair Share (TPS)
 - Belief: most learning happens during the share portion!
 - *Difficult when people don't share*
 - 2 things helping you learn:
 - Try to explain your reasoning
 - Trying to ask a question
- Collaboration, discussion, questions
 - *Slight preference towards less, questions slowing down class*
 - During pauses, the brain learns a lot!

PCM Review: Arrays

`int[]`

type

`arr`

name

=

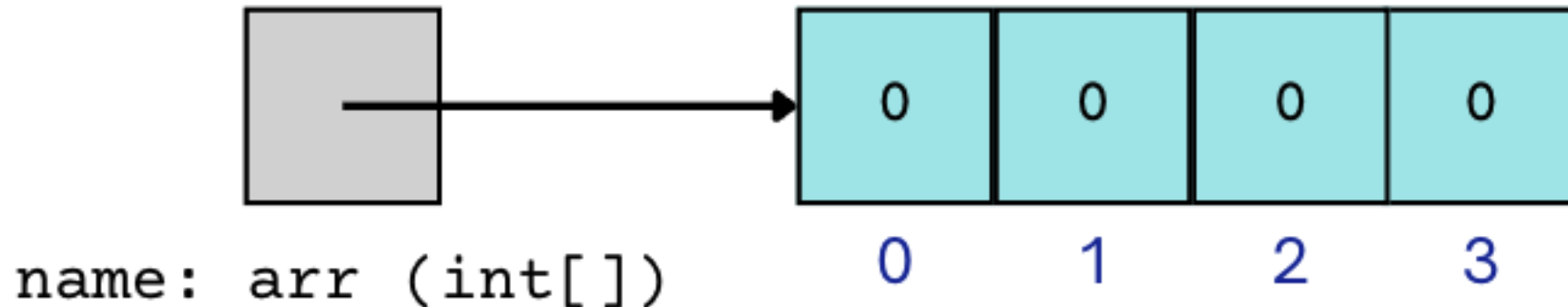
`new int[4];`

array creation code

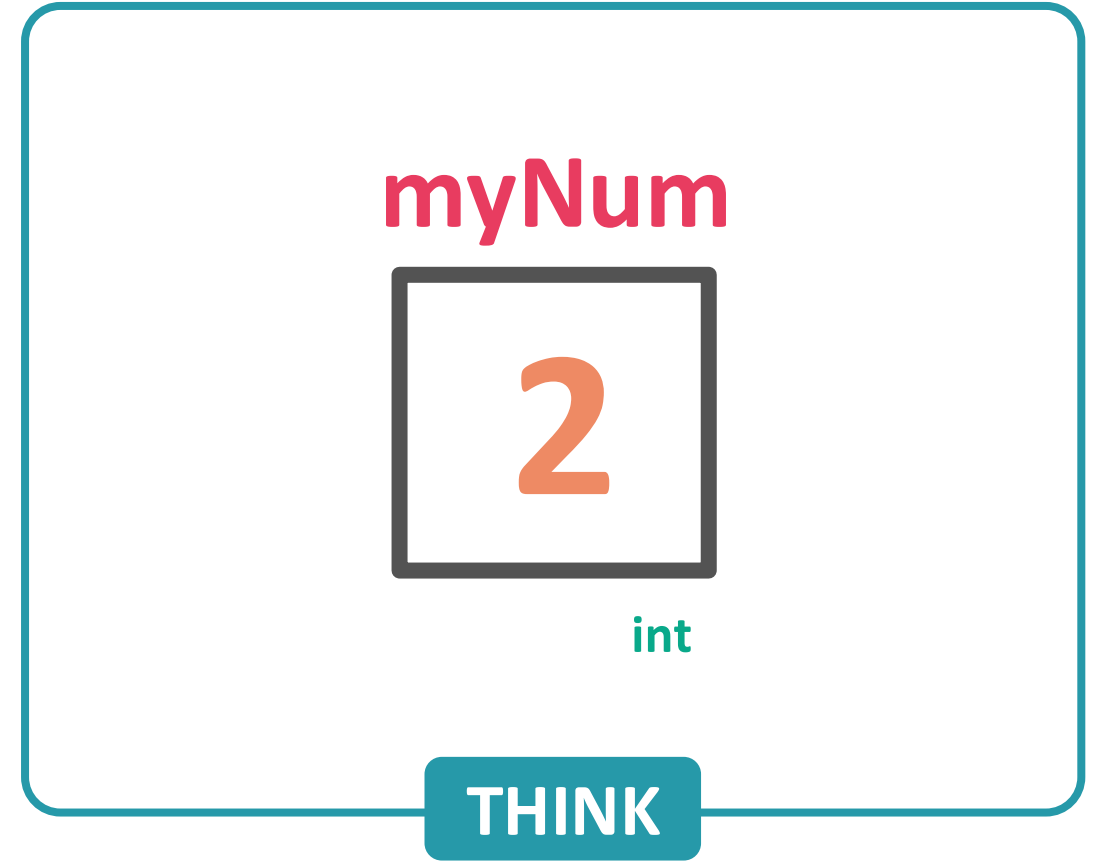
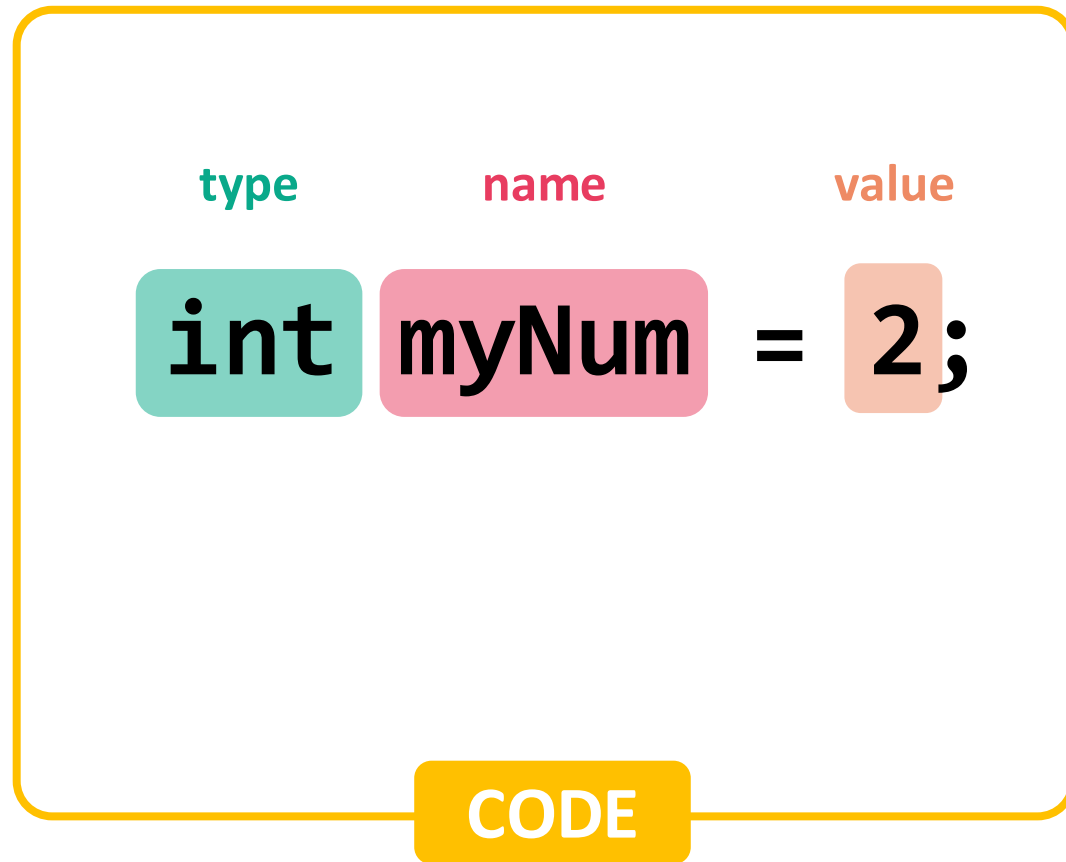
New **data structure** that stores multiple items, in order

- Elements must all be the same type
- Zero-based indexing (like Strings!)

```
int[] arr = new int[4];
```



Recall: Variables and “Mental Models”

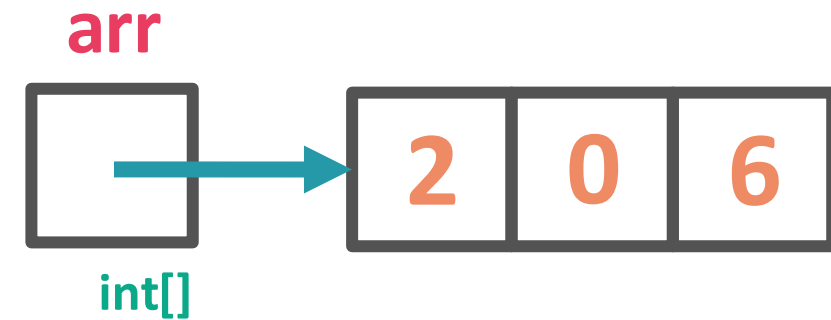


The Mental Model for Arrays

type name value

`int[]` `arr` = `{2, 0, 6};`

CODE



THINK

Arrays vs Strings

Unlike Strings...

- refer to individual elements via bracket notation (e.g. `arr[3]`)
- must decide (and fix) size when created
- `arr.length` to get `arr`'s length
 - *no parentheses*. this is called a “field”, *not* a method.
(more on this in CSE 122!)
- `Arrays.toString(arr)` to get a nice `String` version
 - if not: “gobbledygook” (printing the “reference”)
 - requires `import java.util.*;`

PCM Review: Array Traversal Pattern

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



Practice: Think

[sli.do](#)[#cse121](#)

Which of these expressions gives us the last element of an array `arr`?

- A. `arr[arr.length()]`
- B. `arr[length()]`
- C. `arr[arr.length]`
- D. `arr[arr.length() - 1]`
- E. `arr[arr.length - 1]`



Practice: Pair



sli.do #cse121

Which of these expressions gives us the last element of an array `arr`?

- A. `arr[arr.length()]`
- B. `arr[length()]`
- C. `arr[arr.length]`
- D. `arr[arr.length() - 1]`
- E. `arr[arr.length - 1]`