

Monday February 11

Lecture 11

# Array of Integers (2)

there is pattern on stored values

$$seq[2] = seq[1] + 3$$

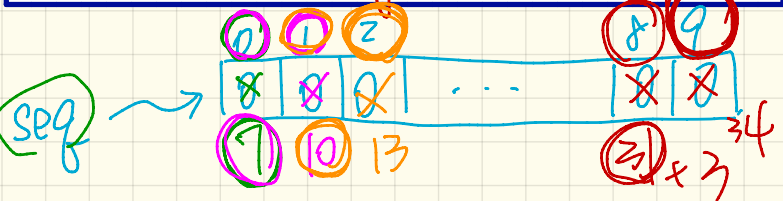
$$seq[1] = seq[0] + 3$$



ia →

## Declaration and Initialization:

```
int[] seq = new int[10];
seq[0] = 7;
for (int i = 1; i < seq.length; i++) {
    seq[i] = seq[i - 1] + 3;
}
```



i	i < seq.length	i-1	seq[i-1]
1	1 < 10 (T)	0	7
2	2 < 10 (T)	1	10
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
9	9 < 10 (T)	8	⋮
10	10 < 10 (F)	9	31

int []

String []

a = new int[?];

names = new - String[?];

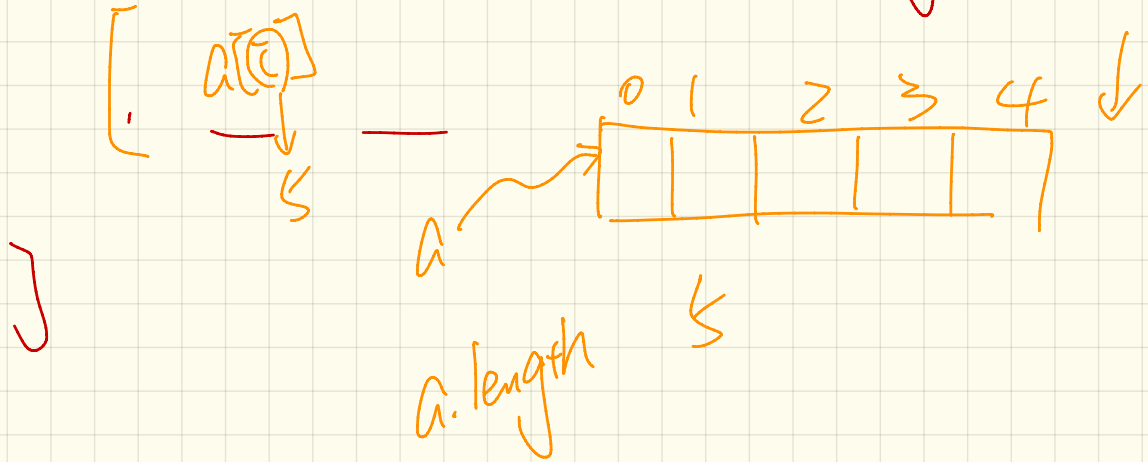
Re-assignment of a slot:

a[2] = 8;

Accessing the value stored in a slot:

println(a[3])

for (int i = 0; i <= names.length; i++) {



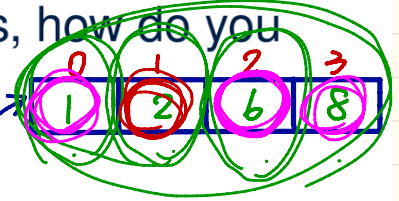
# Computational Problem: Average

**Problem:** Given an array `numbers` of integers, how do you print its average?

$0 < 0$

e.g., Given array `{1, 2, 6, 8}`, print `4.25`.

`numbers`



```
int sum = 0;
for (int i = 0; i < numbers.length; i++) {
    sum += numbers[i];
}
double average = (double) sum / numbers.length;
System.out.println("Average is " + average);
```

$i$	$i < \text{ns.length}$
0	$0 < 4$ ✓
1	$1 < 4$ ✓

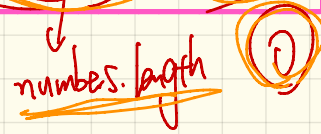
$\text{sum} = \text{sum} + \text{ns}[i]$

Test Case 1  
`int[] numbers = {1, 2, 6, 8}`



Test Case 2  
`int[] numbers = { }`

division by zero



~~17~~  
~~\*~~  
~~\*~~

sum

# Computational Problem: Printing Backwards

**Problem:** Given an array `numbers` of integers, how do you print its contents backwards?

e.g., Given array `{1, 2, 3, 4}`, print 4 3 2 1.



**Solution 1:** Change bounds and updates of loop counter.

```
for(int i = numbers.length - 1; i >= 0; i --) {  
    System.out.println(numbers[i]);  
}
```

Handwritten annotations: `ns[ns.length - 1]` points to `numbers.length - 1`. `ns[2]` is circled in red. `ns[1]` and `ns[0]` are circled in blue. A note says `i + print index => print index = 3 - i`.

**Solution 2:** Change indexing.

```
for(int i = 0; i < numbers.length; i ++){  
    System.out.println(numbers[names.length - i - 1]);  
}
```

Handwritten annotations: `3 - i` points to `names.length - i - 1`. A table shows the mapping of `i` to `print index`:

i	print index
0	3
1	2
2	1
3	0

