LEC 04

CSE 121

for Loops

Questions during Class?

Raise hand or send here

sli.do #cse121

BEFORE WE START

Talk to your neighbors:

What did you do
for the long weekend?

Music: 121 25wi lecture playlist

Instructor: Matt Wang

TAs:

Ailsa	Alice	Chloë	Christopher
Ethan	Hanna	Hannah	Hibbah
Janvi	Judy	Julia	Kelsey
Lucas	Luke	Maitreyi	Merav
Ruslana	Samrutha	Sam	Shayna
Sushma	Vivian		

Announcements, Reminders

- Feedback for C0 released yesterday!
  - please view your feedback – crucial part of learning process
  - for regrades (not resubs), please make a private Ed post
- C1 releasing later today, due Tuesday, Jan 28th
  - first assignment with an "unsolved" problem!
- Resubmission Cycle 0 (R0) opening tomorrow, due Thursday Jan 30th
- Quiz 0 is on Thursday, February 6th (in your registered quiz section)
  - can't make it? email Matt ASAP

Reminder: Resubmissions (or "resubs")

Each week, you may resubmit one Programming Assignment or Creative Project with **no penalty**. The grade of your resubmission will completely replace your previous grade.

This is a huge opportunity: you get to resubmit your work after we grade it and give you feedback! Please take advantage of this :)

(if you miss an assignment and/or only finish it late – use a resub!)

Resub Logistics

Some logistics:

- there are 8 total resub cycles this quarter (and 8 assignments … hm)
- assignments eligible to resubmit for 3 cycles after feedback is out

To resubmit:

1. make and submit your changes
2. set the submission you want graded as “Final”
3. **submit a Google Form**, with a reflection, to confirm your resub
   - you **must** submit the form before the deadline for resub to count

NOTE FOR THOSE AT HOME (A START)

This is one of the rare times where the slides will *differ* for A and B section, since I made an error in the A lecture that I resolved for the B lecture.

The following slides are *only* for those in A section.

(the slides marked “note for those at home” mark the start and end of the A & B lecture slides)

Correcting a mistake…

Last lecture, I did an example out of order – sorry!

Things that I *meant* to explain:

- .toUpperCase()
- single versus double-argument substring
- IndexOutOfBoundsException
- inclusive versus exclusive bounds

I will do that *right now*, but if you were confused on Friday, that's ok!

NOTE FOR THOSE AT HOME (B START)

This is one of the rare times where the slides will *differ* for A and B section, since I made an error in the A lecture that I resolved for the B lecture.

The following slides are *only* for those in B section.

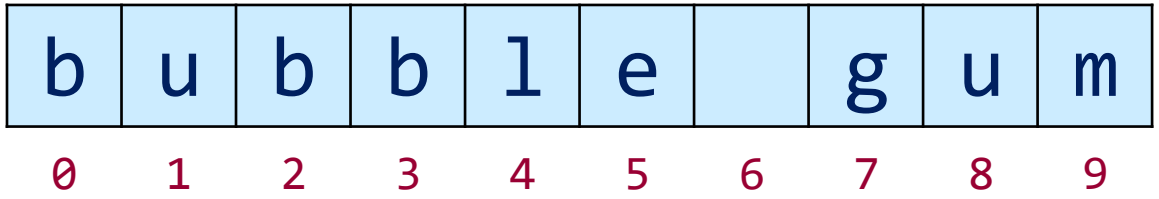
(the slides marked “note for those at home” mark the start and end of the A & B lecture slides)

Practice: Think

sli.do #cse121

Suppose `s` contains the String `"bubble gum"`.

Which statement would result in `s` containing `"Gumball"` instead?



A. `s.substring(7) + "ball";`

B. `s = s.substring(7, 9) + "ball";`

C. `s = s.charAt(7).toUpperCase() + "ball";`

D. `s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";`

Practice: Pair

sli.do #cse121

Suppose `s` contains the String `"bubble gum"`.

Which statement would result in `s` containing `"Gumball"` instead?

b	u	b	b	l	e		g	u	m
0	1	2	3	4	5	6	7	8	9

A. `s.substring(7) + "ball";`

B. `s = s.substring(7, 9) + "ball";`

C. `s = s.charAt(7).toUpperCase() + "ball";`

D. `s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";`

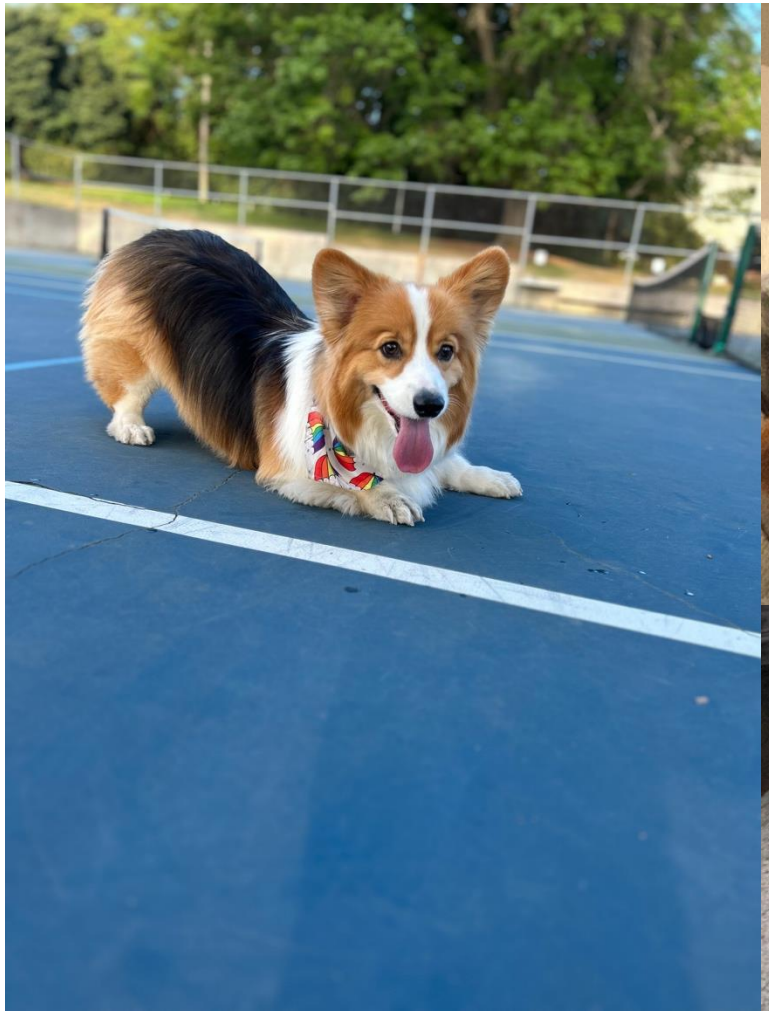
NOTE FOR THOSE AT HOME (A & B END)

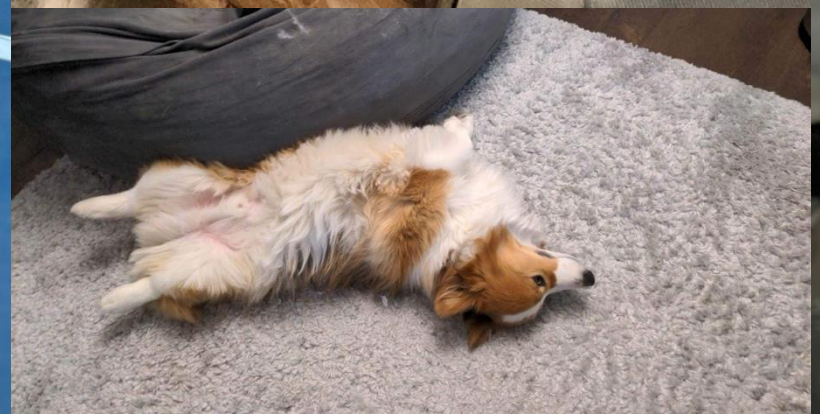
This is one of the rare times where the slides will *differ* for A and B section, since I made an error in the A lecture that I resolved for the B lecture.

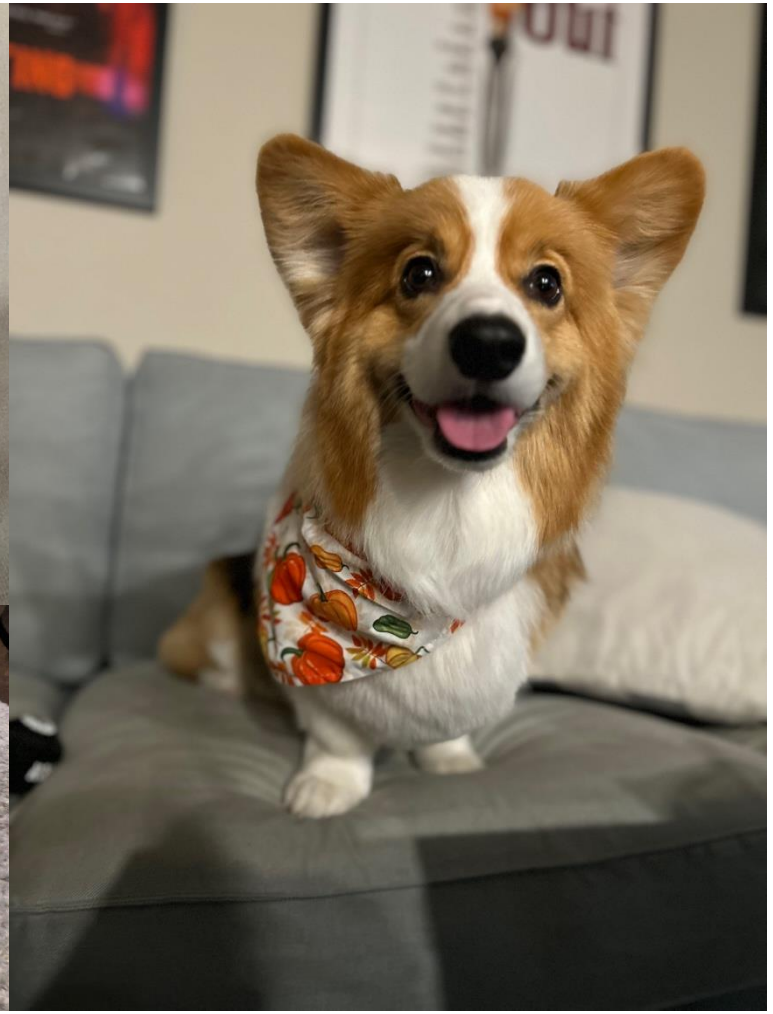
After this slide, we're back to regularly scheduled programming (i.e. the same slides for both).

(the slides marked "note for those at home" mark the start and end of the A & B lecture slides)

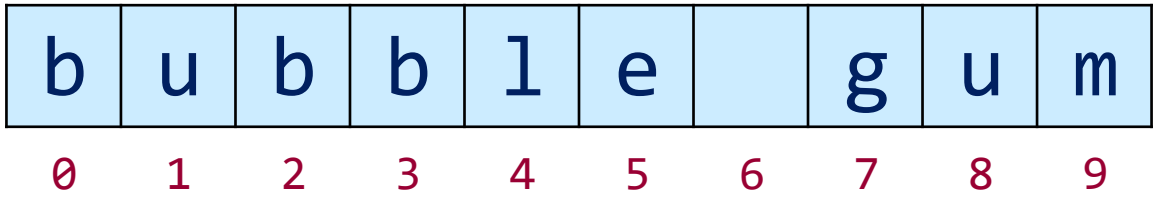
Aside: Gumball







Chaining methods in expressions




```
s.substring(7, 8).toUpperCase() + s.substring(8) + "ball"

"g".toUpperCase() + s.substring(8) + "ball"

"G" + s.substring(8) + "ball"

"G" +      "um"      + "ball"
```

PCM Review: for loops!

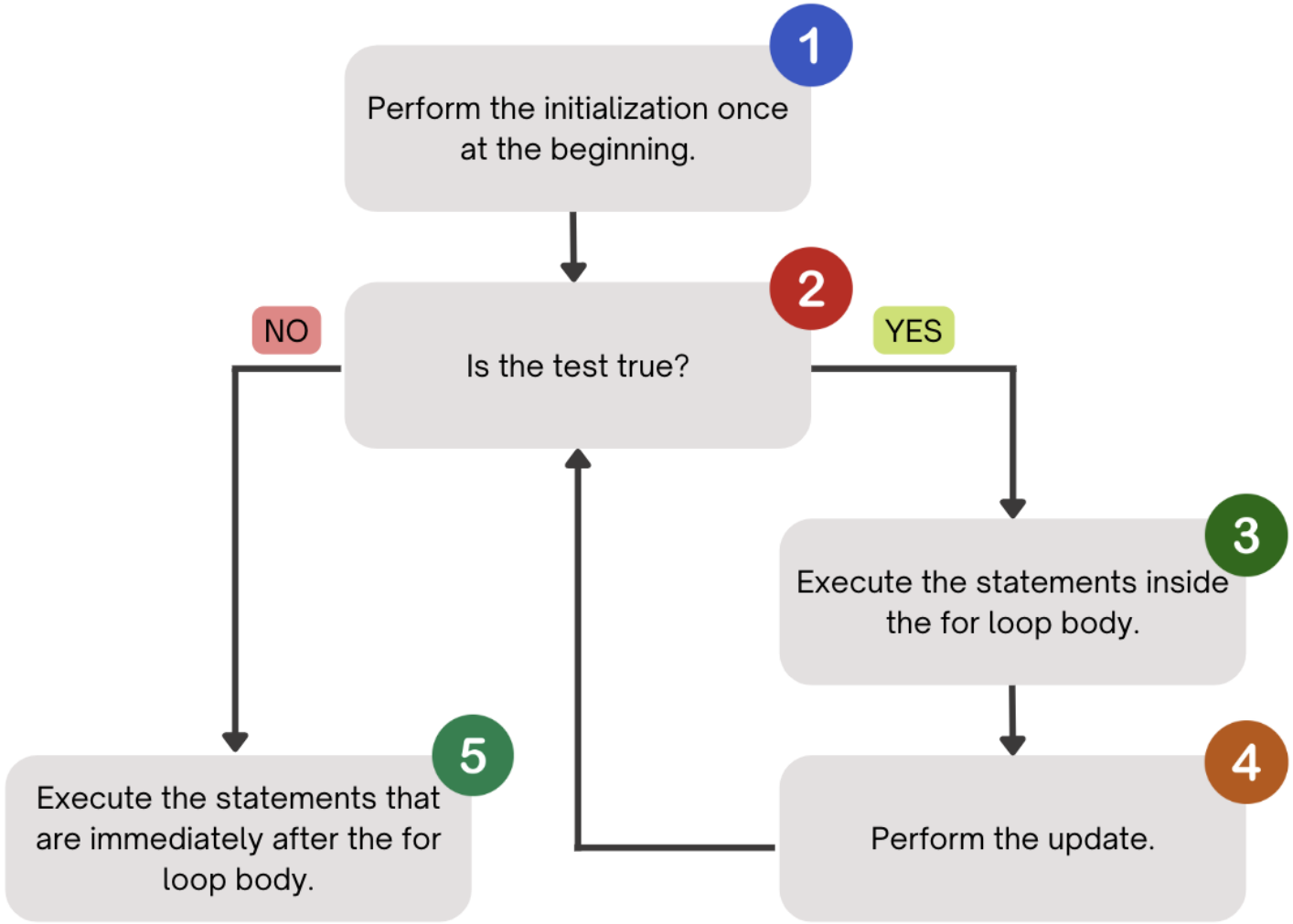
For loops are our first **control structure**: a syntax *structure* that *controls* the execution of other statements.

```
for ( initialization ; test ; update ) {
    body (statements to be repeated)
}
```

PCM Review: for loops (a simple example)

```
for (int counter = 1; counter <= 5; counter++) {
    System.out.println("I love CSE 121!");
}
```

PCM Review: for loops (a helpful flowchart)



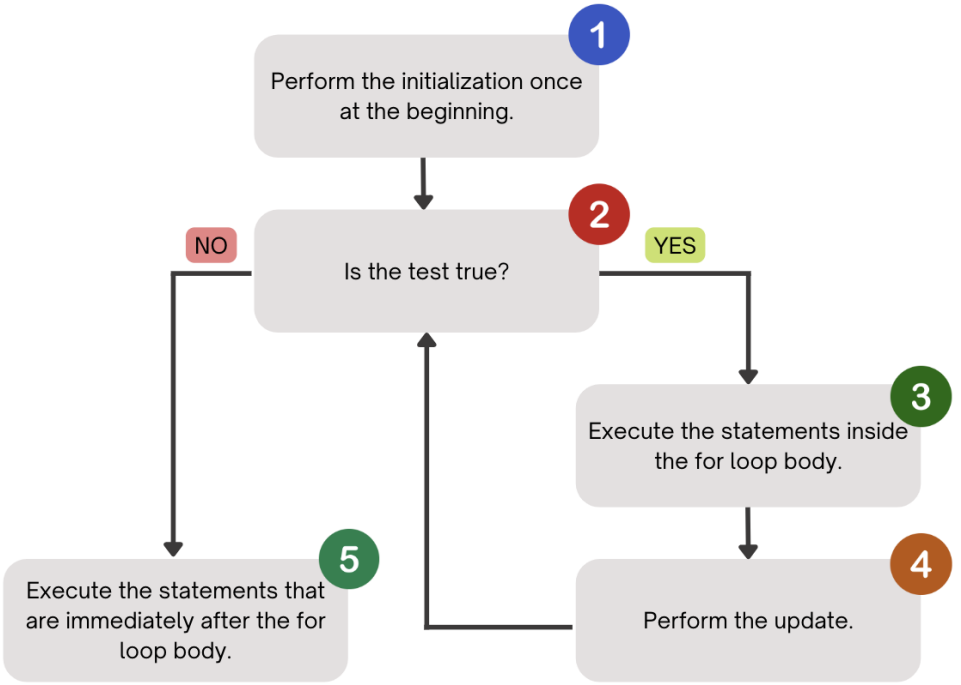
Thinking about for loops

counter

6

```
for (int counter = 1; counter <= 5; counter++) {
    System.out.println("I love CSE 121!");
}
```

```
I love CSE 121!
I love CSE 121!
I love CSE 121!
I love CSE 121!
I love CSE 121!
```



Practice: Think

sli.do #cse121

What output does the following code produce?

```
for (int i = 1; i <= 7; i++) {
  System.out.println(i + " squared = " + i * i);
}
```

A.

```
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
```

B.

```
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
```

C.

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
6 squared = 36
```

D.

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
6 squared = 36
7 squared = 49
```

Practice: Pair

sli.do #cse121

What output does the following code produce?

```
for (int i = 1; i <= 7; i++) {
  System.out.println(i + " squared = " + i * i);
}
```

A.

```
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
```

B.

```
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
i squared = i * i
```

C.

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
6 squared = 36
```

D.

```
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
6 squared = 36
7 squared = 49
```

PCM Review: String Traversals

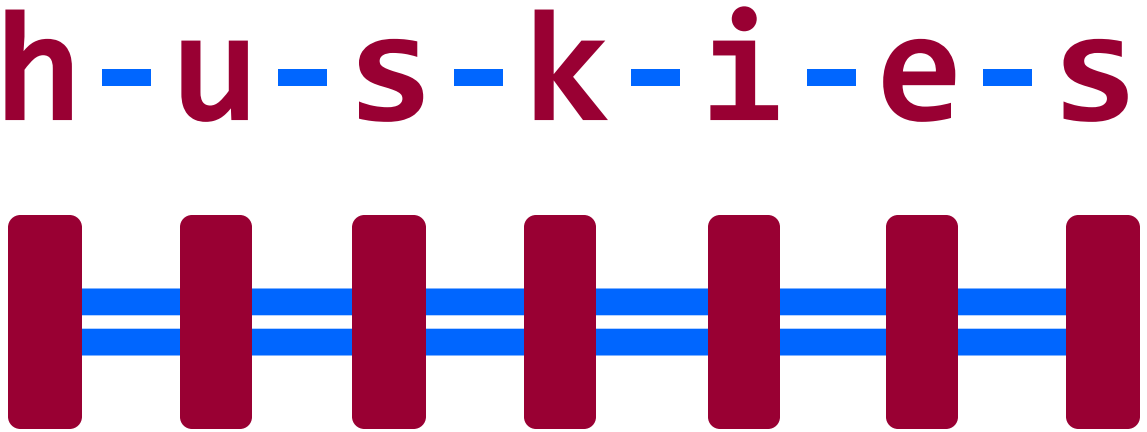
```
// For some String s
for (int i = 0; i < s.length(); i++) {
    // do something with s.charAt(i)
}
```

Go Huskies?

```
h-u-s-k-i-e-s
```

The Fencepost Pattern

Some task where one piece is repeated *n* times, and another piece is repeated *n-1* times and they alternate



Debugging – Live!

In C0 & P0, we asked you to do some debugging.

This is arguably the most important skill when programming – especially because programming is a social activity!

Let's do some live debugging :)

Announcements, Reminders (again)

- Feedback for C0 released yesterday!
  - please view your feedback – crucial part of learning process
  - for regrades (not resubs), please make a private Ed post
- C1 releasing later today, due Tuesday, Jan 28th
  - first assignment with an "unsolved" problem!
- Resubmission Cycle 0 (R0) opening tomorrow, due Thursday Jan 30th
- Quiz 0 is on Thursday, February 6th (in your registered quiz section)
  - can't make it? email Matt ASAP