

LEC 04

CSE 121

for Loops



BEFORE WE START

*Talk to your neighbors:**Have you heard of corgi yoga?*Music: 🌸 [CSE 121 25sp Lecture Tunes](#) 🌸**Instructor:** Miya Natsuhara

TAs:	Chloë	Hibbah	Sushma
	Ailsa	Julia	Kelsey
	Johnathan	Sahej	Shayna
	Christian	Ruslana	Hannah
	Merav	Hanna	Zach
	Judy	Maitreyi	
	Janvi	Ayesha	

Questions during Class?

Raise hand or send here

sli.do #cse121



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
- Grade calculator
- C1 releasing later today, due Tuesday, April 22nd
- Quiz 0 is on Thursday, April 24th (in your registered quiz section)
 - Can't make it? Email Miya ASAP
- Resubmission Cycle 0 (R0) opening tomorrow, due Thursday April 24th



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)
- 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



Last Time...

- Variables
 - Container that stores a specific data type
 - Must declare and initialize
 - `int version = 5;`
- Strings
 - Sequence of characters treated as one, yet can be indexed as individual parts
 - `char` represents a single character

version

5

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





Practice: Think

[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";`





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";`



Chaining methods in expressions

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

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

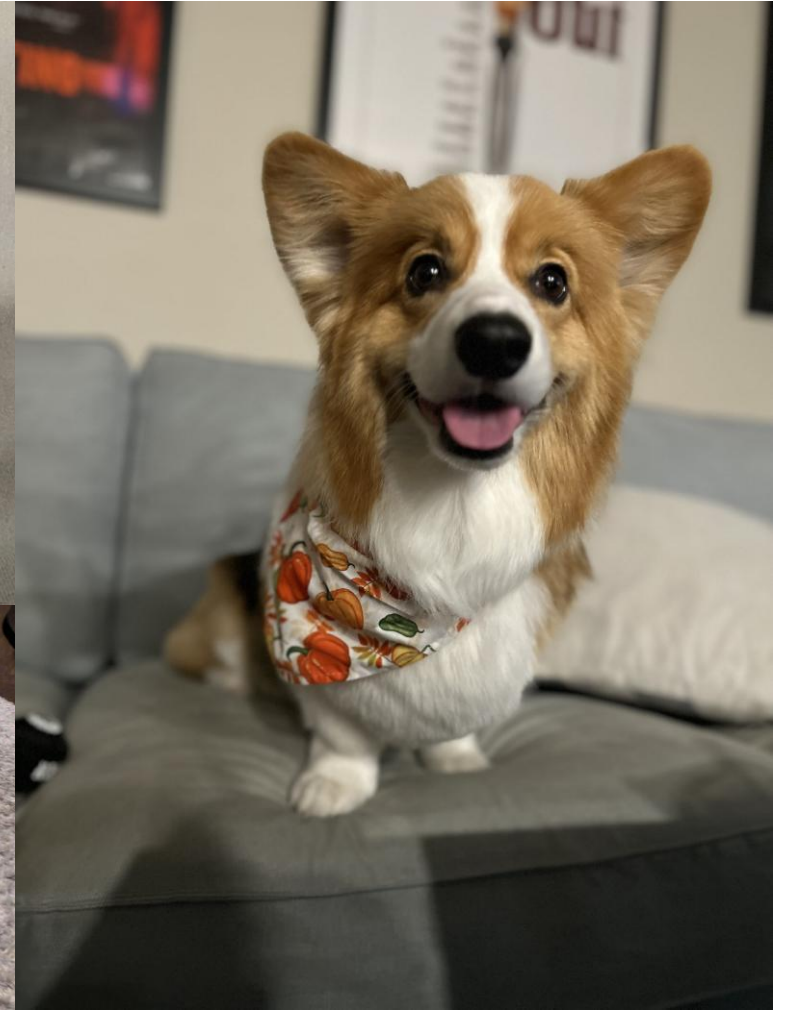
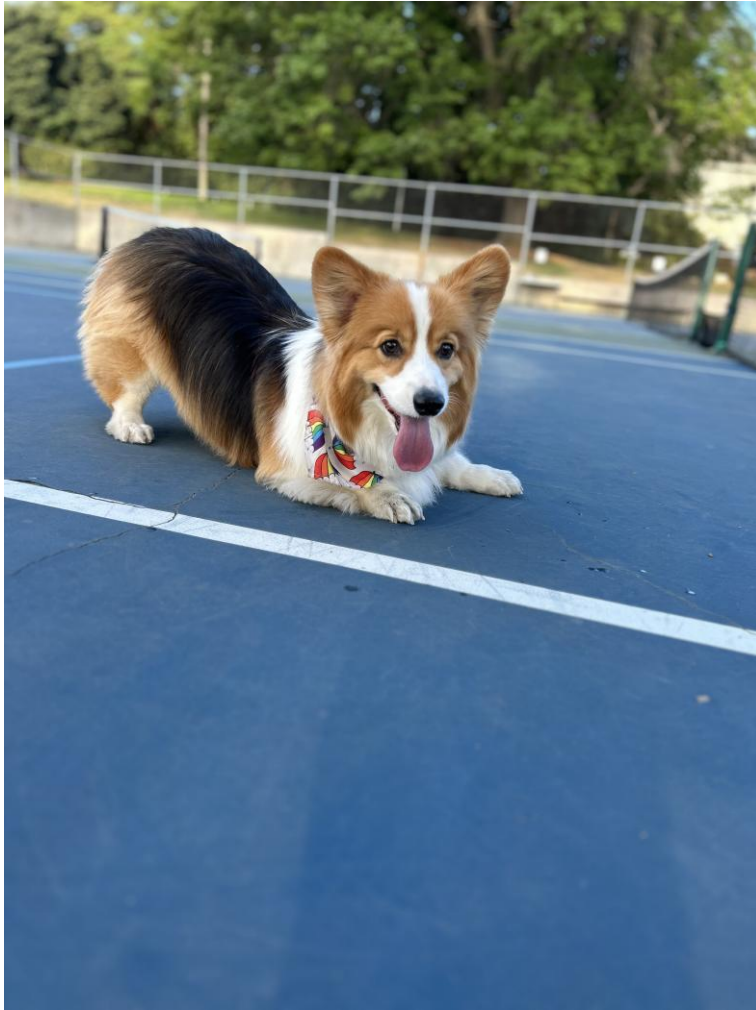
```
"g".toUpperCase() + s.substring(8) + "ball"
```

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

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



Aside: Gumball



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)  
}
```

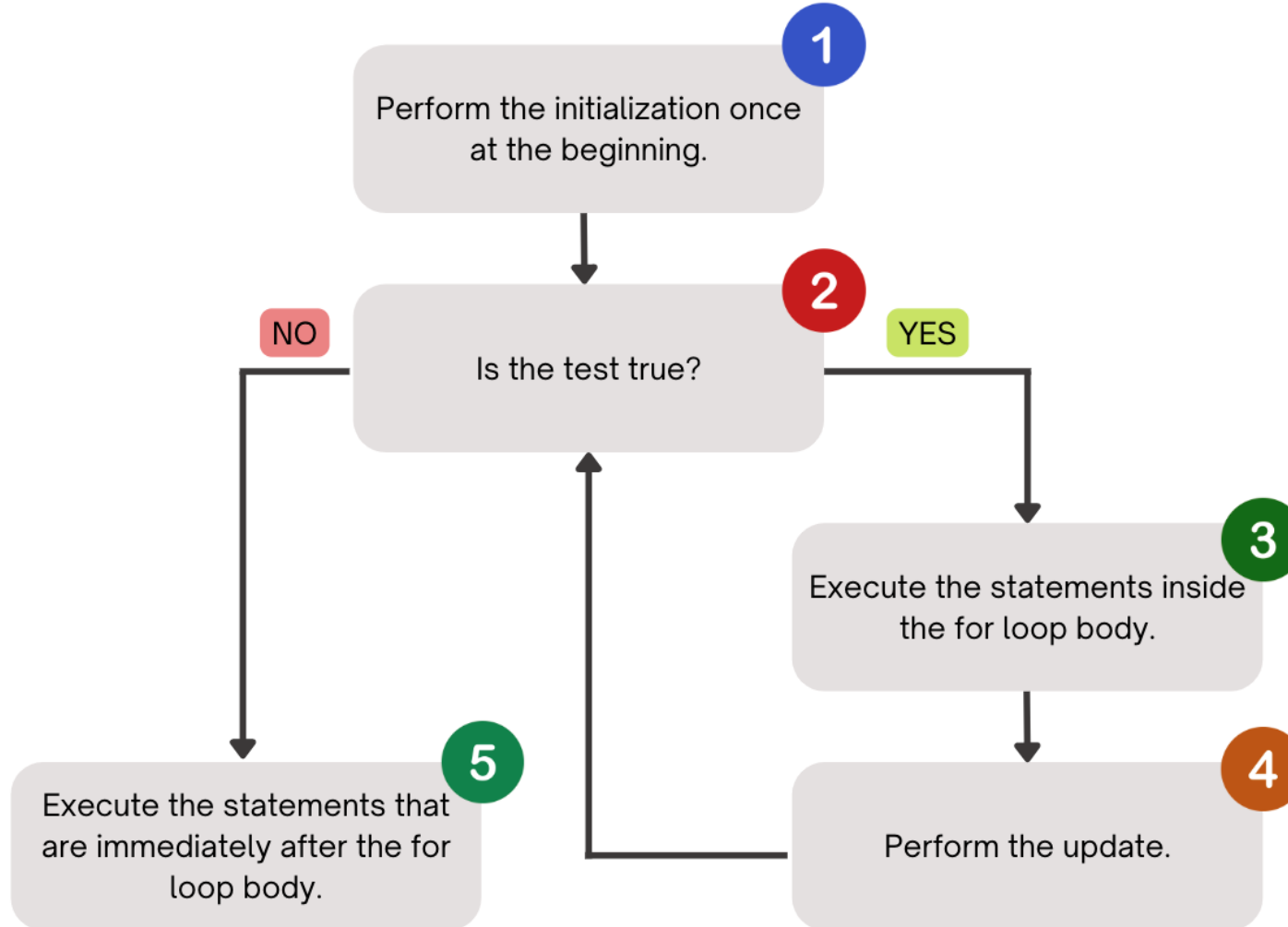


PCM Review: for loops (example)

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



PCM Review: for loops (a helpful flowchart)



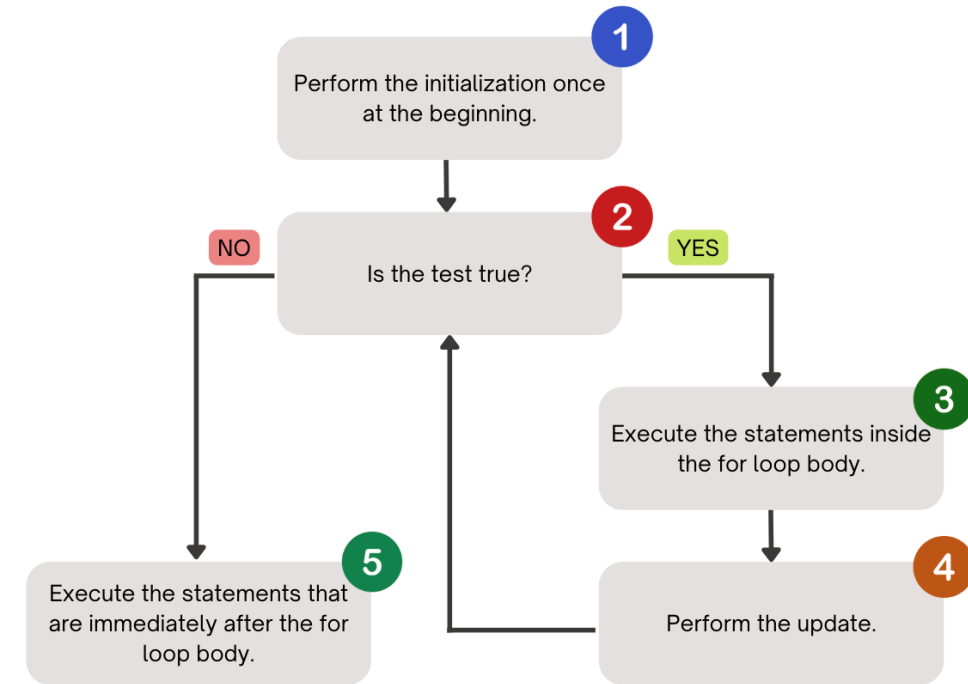
Thinking about for loops

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

counter

6

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)  
}
```

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



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

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

