

LEC 05

CSE 121

for Loops

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

*Talk to your neighbors:**What's your favorite TV show right now?*Respond on [sli.do!](#)Music: [🌿 CSE 121 26au Lecture Tunes 🌿](#)**Instructor:** Miya Natsuhara and Anjali Pal

TAs:	Samrutha	Tamsyn	Owen	Jennie
	William	Cedric	Jameson	Anisha
	Samvedha	Anum	Paul	TJ
	Syreeta	Navya	Gurbax	Ivory
	Ryan	Gene	Tony	Sarah W
	Anant	Margot	Lucas	Avery
	Henry	Sarah K	Alex	
	Reese	Ethan	Jesse	
	Savannah	Evelyn	Matthew	

Announcements, Reminders

- Computer Quiz 0 window currently ongoing!
 - Make sure to sign up for a timeslot if you have not already!
 - Window ends on Wed 10/21 – if you wait until the last minute, there may not be any timeslots left
- Paper Quiz 0 is *next* Tuesday, October 20 (in your registered quiz section)
 - Can't make it? Email ASAP (cse121-instructors@cs.washington.edu)
 - [Paper Quiz Logistics announcement](#)
- Section Participation grades for Section 0, 1, 2 posted on Canvas!
 - Reach out to your quiz section TA if something looks off

PCM Review: for loops!

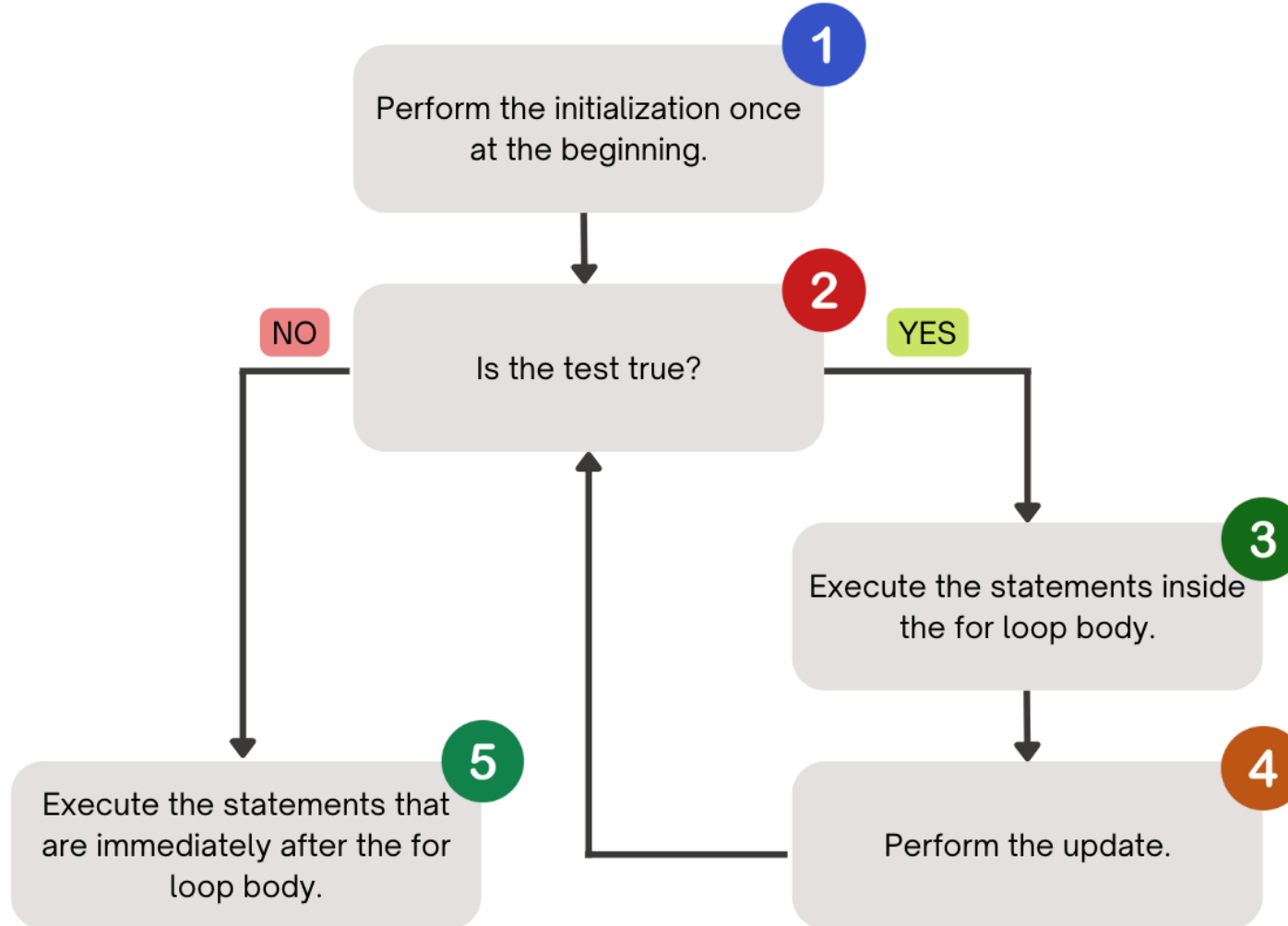
For loops are our first **control structure**: a syntax *structure* that *controls* the execution of other statements.

```
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)



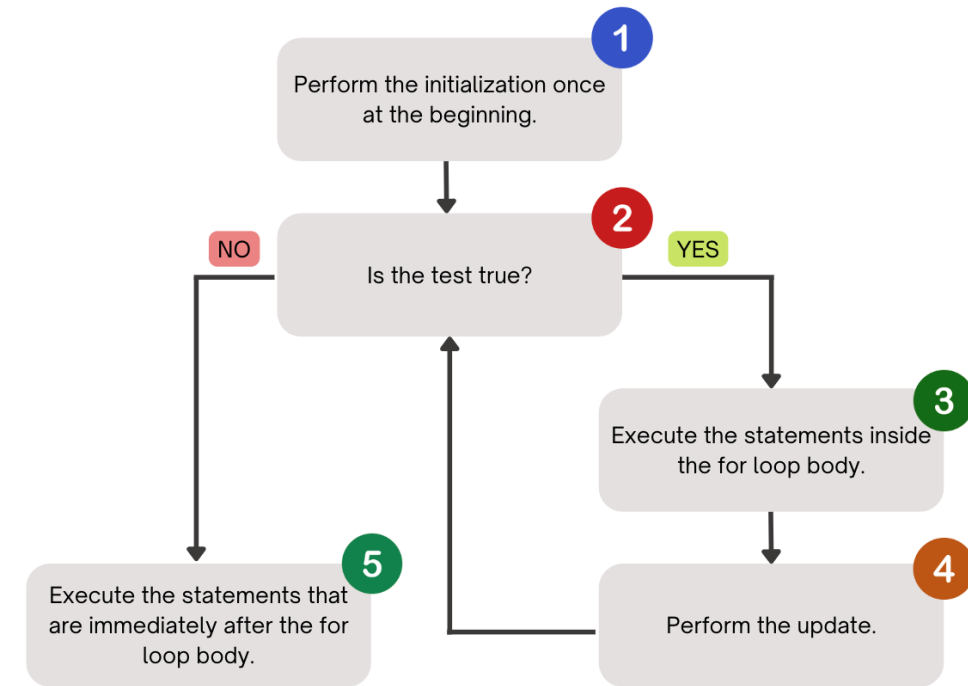
Tracing through a for loop

```
1 for (int counter = 1; counter <= 5; counter++) {  
2   3 System.out.println("I love CSE 121!");  
4 }  
5
```

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
```

PCM: Math

Math.<method>(...)

Method	Description
<code>Math.abs(<i>value</i>)</code>	Returns the absolute value of <i>value</i>
<code>Math.ceil(<i>value</i>)</code>	Returns <i>value</i> rounded up
<code>Math.floor(<i>value</i>)</code>	Returns <i>value</i> rounded down
<code>Math.max(<i>value1</i>, <i>value2</i>)</code>	Returns the larger of the two values
<code>Math.min(<i>value1</i>, <i>value2</i>)</code>	Returns the smaller of the two values
<code>Math.round(<i>value</i>)</code>	Returns <i>value</i> rounded to the nearest whole number* note: need to cast result to int (it's complicated!)
<code>Math.sqrt(<i>value</i>)</code>	Returns the square root of <i>value</i>
<code>Math.pow(<i>base</i>, <i>exp</i>)</code>	Returns <i>base</i> raised to the <i>exp</i> power

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

