

# CSE 121 – Lesson 2

Kai Daniels

Summer 2023

Music:  [k-pop girlies playlist](#) 



[sli.do #cse121](https://sli.do/#cse121)


# Announcements, Reminders

- Programming Assignment 0: Code Reviews was released Wed, due next Wed July 5<sup>th</sup> 11:59 PM
- Creative Project 0 feedback released next Tues, July 4
- Quiz 0: Monday, July 10 (take-home)
- Resubmission start next week, more on that later


# Last Time...

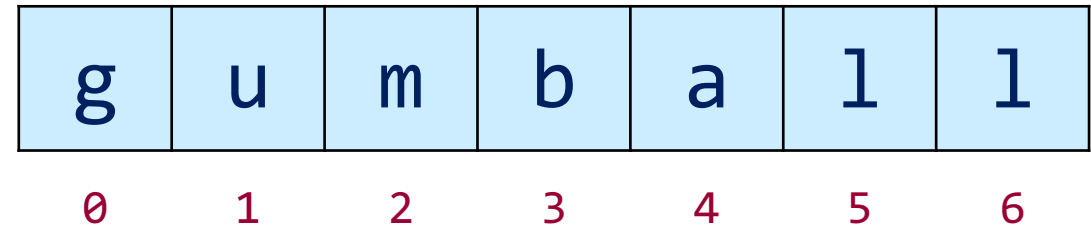
- Expressions
  - Data types (int, double, boolean, String)
  - Operators (+, -, /, \*, %)
  - Logical Operators (!, &&, ||, ==, !=, >, >=, <, <=)

```
// declare AND initialize  
int version = 5;
```

- Variables
  - Container that stores a specific data type
  - Must declare & initialize!
  - Manipulate, modify, reuse 

# (PCM) Strings and chars

- String = sequence of characters treated as one, yet can be indexed to get individual parts
- Zero-based indexing 



- **Side note:** new data type!  
char, represents a single character,  
so we use single quotes  
Strings are made up of chars!

# (PCM) String Methods

Usage: `<string variable>.<method>(…)`

Method	Description
<code>length()</code>	Returns the length of the string.
<code>charAt(i)</code>	Returns the character at index <i>i</i> of the string
<code>indexOf(s)</code>	Returns the index of the first occurrence of <i>s</i> in the string; returns -1 if <i>s</i> doesn't appear in the string
<code>substring(i, j)</code> or <code>substring(i)</code>	Returns the characters in this string from <i>i</i> (inclusive) to <i>j</i> (exclusive); if <i>j</i> is omitted, goes until the end of the string
<code>contains(s)</code>	Returns whether or not the string contains <i>s</i>
<code>equals(s)</code>	Returns whether or not the string is equal to <i>s</i> (case-sensitive)
<code>equalsIgnoreCase(s)</code>	Returns whether or not the string is equal to <i>s</i> ignoring case
<code>toUpperCase()</code>	Returns an uppercase version of the string
<code>toLowerCase()</code>	Returns a lowercase version of the string

# Poll in with your answer!



Suppose `s` contains the String "bubble gum". Which option below would result in `s` containing "Gumball" instead?

```
String s = "bubble gum";
```

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";`
- E. `s = s.substring(7, 8).toUpperCase() + s.substring(7, 10) + "ball";`

# (PCM) for loops!

For loops are our first *control structure*

A syntactic structure that *controls* the execution of other statements.

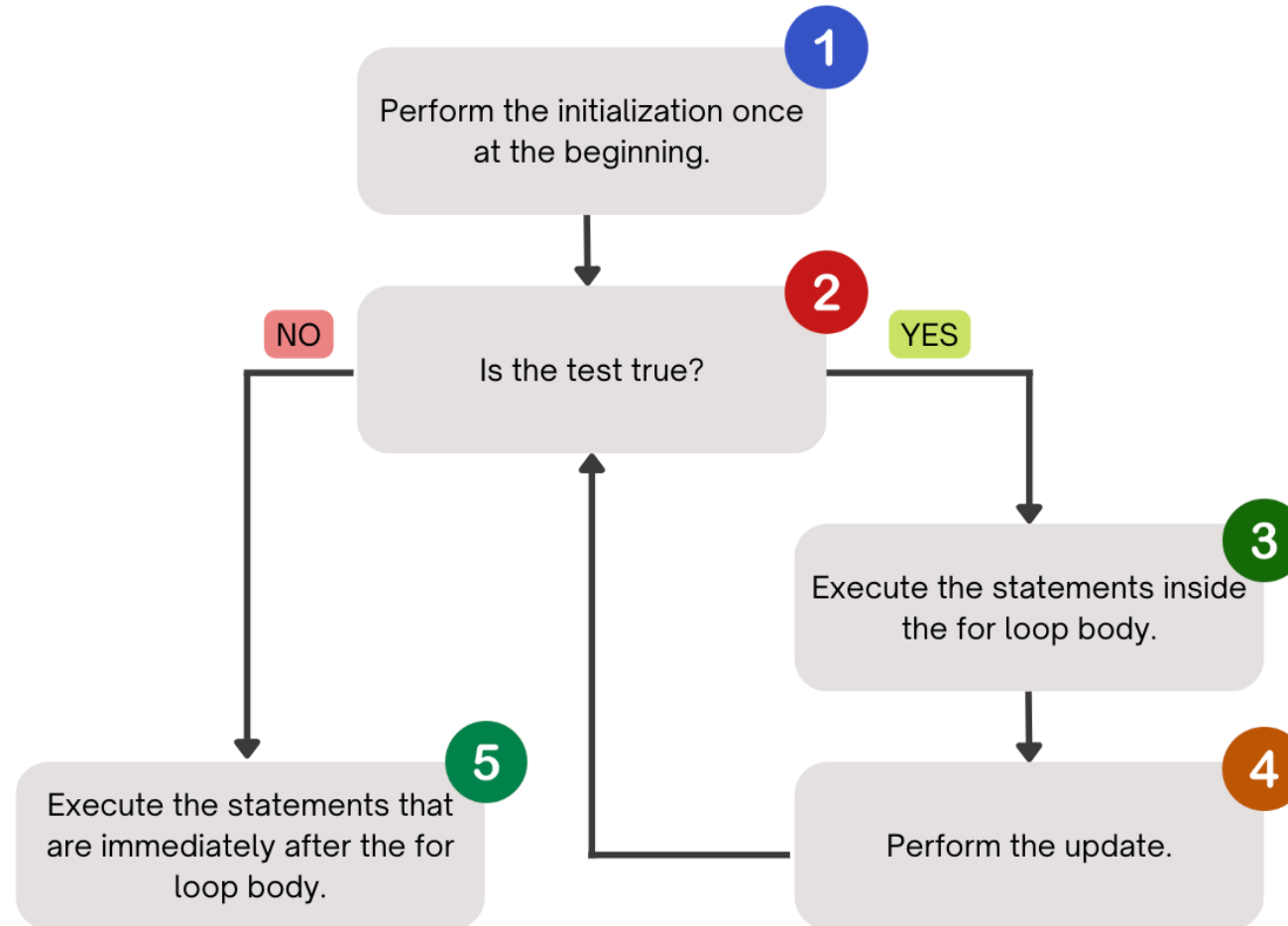
```
for ( initialization ; test ; update ) {  
    body (statements to be repeated)  
}
```

# (PCM) for loops!

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



# (PCM) for loops!



# Poll in with your answer!



What output does the following code produce?

```
for (int i = 1; i <= 6; 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
```

C.

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

D.

```
1 squared = 11  
2 squared = 22  
3 squared = 33  
4 squared = 44  
5 squared = 55  
6 squared = 66  
7 squared = 77
```

# (PCM) String traversals

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

# Fencepost Pattern

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

**g-u-m-b-a-1-1**

# Fencepost Pattern

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

