

LEC 07

CSE 121

Methods, Parameters, Scope (revisited)

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

*Talk to your neighbors:**Is cereal a soup? Why?*

Instructor: Hayden Feeney**TAs:** Anisha Savannah William
Cayden Tamsyn

Agenda (1/4)

- **Announcements, Reminders**
- Closing the loop on C1
- Methods & Parameters Review
- Programming with Methods & Parameters

Announcements, Reminders

- **P1** is out, due next Tuesday, July 21st
 - start early!!
- **R1** released yesterday, due Thursday, July 23rd
- **Quiz 0** was yesterday
 - grades will be out before Quiz 1

Agenda (2/4)

- Announcements, Reminders
- **Closing the loop on C1**
- Methods & Parameters Review
- Programming with Methods & Parameters

ASCII Art: part 1!

Task 1: A
small
spaceship

```

  /\
 /__\
|oo|
/||\
o00o

```

Task 1: One
Robot

```

[oo]
/||\
 /\

```

Task 1: A piece of
flan!

```

  ____
 /    \
/      \
/____\

```

Task 1: a
little tree

```

  /\
 /  \
/____\
 ||

```

Task 1: A 4-
Star Dragon
Ball!

```

  ____
 /  *  *  \
|    *  *   |
 \_____/

```

Task 1: Single
shulker box from
Minecraft!

```

-----
|      |
|      |
|  _  |
|  -  |
|      |
-----

```

Task 1: A cat

```

      o      o
    ooo      ooo
  oooooooooooooooooo
 oooooooooooooooooooo
oooo 0 ooo 0 ooo
  oooooo ^ ooooo
    ooooooooooooo

```

Task 3: a four caterpillar dance
troupe!

```

  ' _ '      ' _ '      ' _ '      ' _ '
(. _ .)/  (. _ .)/  (. _ .)/  (. _ .)/
<( )      <( )      <( )      <( )
( )        ( )        ( )        ( )
<__>      <__>      <__>      <__>

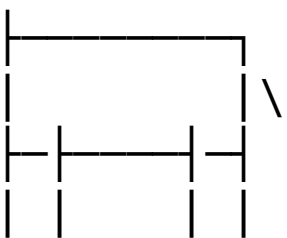
```

ASCII Art: part 2!

Task 1: A heart using
< and 3, stack
together to create a
bigger heart

```
<3<3  <3<3
<3<3<3<3<3<3
  <3<3<3<3
    <3<3
```

Task 1: 0x
LoJ



Task 1: A music
half note.



Task 1: The image is an ASCII portrayal of an old-school computer monitor displaying the classic video game, Pong. The computer monitor has a thick 3D outer border and is shown under a title banner displaying the word "Pong". The inner screen displays opposing paddles on opposite sides and a ball bouncing between them.

The image is displayed below:

```

~ P   O   N   G ~
.-----|.
|  \IIIIIIIIIIII/  |
|  |      :      |  |
|  |      :      |  |
|  |      :  o    |  |
|  |      :      |  |
|  |      :  [    |  |
|  |      :      |  |
|  /IIIIIIIIIIII\  |
|-----|.

```

Closing Loop on C1's Reflection – ASCII alts

- Class consensus: **writing alt text is hard!**
 - generally: hard to know what to include!
balance being concise with being descriptive enough.
 - what if the alt text relies on knowing what something “looks like”?
 - (also: what if your *grader* needed alt text?)
- nobody is born knowing how to write alt text; **it takes practice!**
 - for example: the course staff is learning to write alt text too!
 - want to learn more? check out optional resources in C1's reflection!

Closing Loop on C1's Reflection – Q2

Q2: [...] Jake discusses how deciding on alt text for other people's headshots made him feel "judgey". Specifically, he says:

Because I was looking at people's avatars, and going, you do not deserve an alt, because it's just you. Like, you're not doing a thing, it's just a head shot or whatever.

The 121 course staff disagrees with his approach to writing alt text for others, particularly that some headshots do not "deserve" alt text.

Closing Loop on C1's Reflection – Q2 answers! (1)

- “I think his take in saying this was that some images are self explanatory, and altatts should only be used in order to describe emotion that isn't visible to a reader. However, I think that this process could go wrong because the reader still wouldn't be able to know what exactly the image is, so the images with alt text would be the only images viewed by people with disabilities, and the ones without any altatts wouldn't be viewed by people with disabilities. So it creates another barrier in terms of online accessibility when viewing images. This can be avoided by adding a simple alt attribute like "This is so and so posing for a headshot" or something that describes the image in a concise way.”

Closing Loop on C1's Reflection – Q2 answers! (2)

- “This situation could go unintentionally wrong because if the audience has no alt text available, only Jake's judgment of the people he deems worthy of an alt will be passed on... As a result, Jake's judgments about everyone will also become the judgments of everyone relying on the alts.”

How to avoid this harm?

- Give everyone alt text
- *Attempt* to be neutral and respectful

Closing Loop on C1's Reflection – alts IRL

We asked you where alt text might appear in your life.
Many great answers!

- on Instagram ([Instagram supports alt text for photos](#))
- shopping sites (Target, Amazon)
- many others (Tumblr, Pinterest, League of Legends, Reddit, memes!)
- and in different contexts (classes, jobs, RSOs, etc.)

Alt text is not *always* right!

- in many cases, concise alt text *is not enough*
 - for information-dense content: GIS maps, video for an RSO, a complicated physics graph, or a flowchart
 - for *interactive* content: video games, Solidworks, Figma
- so, there are alternatives
 - by adding more structure: captions, tables, or web pages
 - by using audio (e.g., “[audio descriptions](#)”, “[sonification](#)”)
 - by creating alternate interaction techniques
- this topic will return... in C3!

Agenda (3/4)

- Announcements, Reminders
- Closing the loop on C1
- **Methods & Parameters Review**
- Programming with Methods & Parameters

PCM: Methods

Writing our own **methods** allows us to define our own commands!
(naming conventions for methods are same as variables: camelCased)

```
public static void myMethod() {  
    /**  
    Your code here  
    **/  
}
```

This method could then be called like so: `myMethod()`;



Practice: Think



sli.do #cse121

```
public class HelloGoodbye {  
    public static void main(String[] args) {  
        welcome();  
        hello();  
        goodbye();  
    }  
  
    public static void hello() {  
        System.out.print("Hello! ");  
        glad();  
    }  
  
    public static void goodbye() {  
        System.out.println("Goodbye!");  
    }  
  
    public static void welcome() {  
        System.out.print("Welcome! ");  
        glad();  
    }  
  
    public static void glad() {  
        System.out.println("Glad you're here.");  
    }  
}
```

What is the output of this program?

A.

Welcome! Glad you're here.
Hello! Glad you're here.
Goodbye!

C.

Welcome! Hello! Goodbye!

B.

Welcome!
Hello!
Goodbye!

D.

Welcome!
Glad you're here.
Hello!
Glad you're here.
Goodbye!

PCM: Parameters

Definition: a value passed to a method by its caller. “Like” a variable!

```
public static void myMethod(String musicalAct) {  
    System.out.print(musicalAct + " is the best!");  
    ...  
}
```

Calling a method with a parameter...

```
myMethod("Laufey"); // prints: Laufey is the best!
```


PCM: Scope, Redux

Our scope rules also apply to methods and parameters!

- general rule: from its **declaration to the next closing brace, }**
- a variable declared in a method only exists in that method!

```
public static void example(int n) {  
    System.out.println("hello");  
    int x = 3;  
    for (int i = 1; i <= n; i++) {  
        System.out.print(x);  
    }  
}
```

i's scope {

x's scope {

n's scope {



Practice: Think

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

```
public static final int COUNT = 7;

public static void main(String[] args) {
    int count = 5;
    line(count);
    System.out.println("COUNT is: " + COUNT);
    System.out.println("count is: " + count);
}

public static void line(int count) {
    for (int i = 1; i <= count; i++) {
        System.out.print("*");
    }
    count++;
    System.out.println();
}
```

What will be the **last line of output** from this code?

A. count is: 1

B. count is: 5

C. count is: 6

D. count is: 7

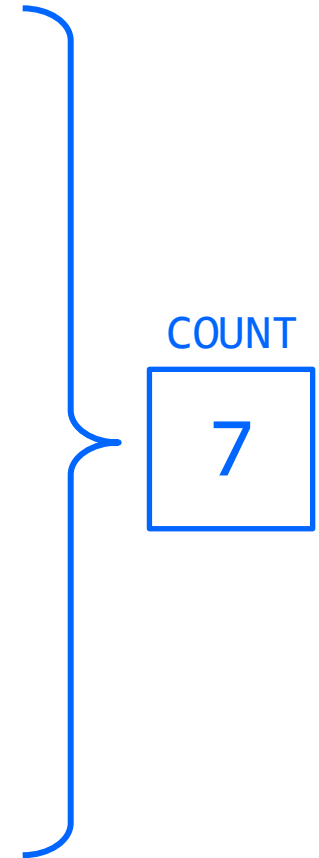
Walkthrough: Counting Counts



```
public static final int COUNT = 7;  
public static void main(String[] args) {  
    int count = 5;  
    line(count);  
    System.out.println("COUNT is: " + COUNT);  
    System.out.println("count is: " + count);  
}
```



```
public static void line(int count) {  
    for (int i = 1; i <= count; i++) {  
        System.out.print("*");  
    }  
    count++;  
    System.out.println();  
}
```



Agenda (4/4)

- Announcements, Reminders
- Closing the loop on C1
- Methods & Parameters Review
- **Programming with Methods & Parameters**

PCM: *Refactoring* Line

```
public static void lineOf5() {  
    for (int i = 1; i <= 5; i++) {  
        System.out.print("*");  
    }  
}  
  
public static void lineOf10() {  
    for (int i = 1; i <= 10; i++) {  
        System.out.print("*");  
    }  
}
```



```
line(5);  
line(10);  
  
public static void line(int numStars) {  
    for (int i = 1; i <= numStars; i++) {  
        System.out.print("*");  
    }  
}
```

New: Method Comments

Each method you write (except main) should have a short comment!

```
// Takes in a Random object as a parameter.  
// Uses the Random object to generate an addition problem  
// where the operands are in the range 1-10 (inclusive),  
// and prints the result, rounded to two decimal places.
```

```
public static void addTwoRandomNumbers(Random randy) {  
    int num1 = randy.nextInt(10) + 1;  
    int num2 = randy.nextInt(10) + 1;  
    int sum = num1 + num2;  
    ...  
}
```