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?

Music: CSE 121 26sp Lecture Tunes

Instructor: Matt Wang

TAs:

Abdul	Amogh	Anant	Anum	Cayden
Dalton	Ethan	Hayden	Jesse	Jessica
Johnathan	Minh	Navya	Paul	Reese
Ruslana	Sam	Savannah	Spencer	Shayna
Tamsyn	TJ	Trey		

Agenda

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

Announcements, Reminders

- **P1** is out, due next Tuesday, April 28th
  - start early!!
- **R1** released yesterday, due Thursday, April 30th
- **Quiz 0** was yesterday
  - grades will be out before Quiz 1

Awesome ASCII Art: Diverse Drawings

Task 1: ASCII art of a tree

```
   ^
  ^^^
 ^^^^^
^^^^^^^
  [ ]
```

Task 1: ASCII art of a lighthouse in the sea.

```
~~~~/\~~~~
~~~/==\~~~
~~~|[]|~~~
~~=\\===~~
~~==\\==~~
~~===\\=~~
~~\===\\~~
```

Task 1: I drew a picture of a planet using the basic println method.

```
  * ._------_ +
    -==_ _  -=_ .
 . ===--  ---===
   _==  ___--_-
*    --_____--  .
```

Task 1: This is a Moorish idol, which is a tropical fish.

```
        *
         *
       * *
*    * * * *
 * * * * * 0 *
*    * * * *
       * *
         *
        *
```

Task 1: A tabby cat!

```
  ^  ||| ^
~(  ^ * ^  )~
( _  o  o _ )
```

Task 1: one baby penguin facing right

```
 __
(  0>
/   \
| V  |
\   /
 ---=
```

Task 1: pig peaking over its pen!

```
 ^ -- ^
 (*  *)
|------|
```

Task 1: A face that looks hauntingly like Voldemort.

```
  ________
 /        \
/ ___)(___ \
| <O>||<O> |
|   /..\   |
 \  ____  /
  \______/
```

Task 1: A donut!

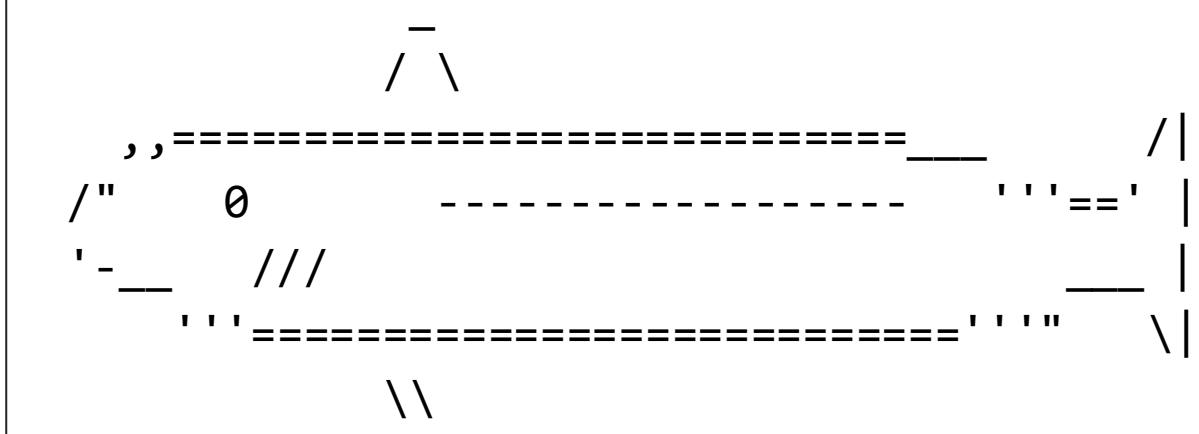
```
     ********
  **************
*******    *******
******      ******
*******    *******
  **************
     ********
```

Awesome ASCII Art: Square Dancing

```
Creative Option 1: 25 diamond men square dancing.
     /\        /\        /\        /\        /\
    /MM\      /MM\      /MM\      /MM\      /MM\
   / .. \    / .. \    / .. \    / .. \    / .. \
   \ -- /    \ -- /    \ -- /    \ -- /    \ -- /
    \  /      \  /      \  /      \  /      \  /
     \/        \/        \/        \/        \/
     /\        /\        /\        /\        /\
    /MM\      /MM\      /MM\      /MM\      /MM\
   / .. \    / .. \    / .. \    / .. \    / .. \
   \ -- /    \ -- /    \ -- /    \ -- /    \ -- /
    \  /      \  /      \  /      \  /      \  /
     \/        \/        \/        \/        \/
     /\        /\        /\        /\        /\
    /MM\      /MM\      /MM\      /MM\      /MM\
   / .. \    / .. \    / .. \    / .. \    / .. \
   \ -- /    \ -- /    \ -- /    \ -- /    \ -- /
    \  /      \  /      \  /      \  /      \  /
     \/        \/        \/        \/        \/
     /\        /\        /\        /\        /\
    /MM\      /MM\      /MM\      /MM\      /MM\
   / .. \    / .. \    / .. \    / .. \    / .. \
   \ -- /    \ -- /    \ -- /    \ -- /    \ -- /
    \  /      \  /      \  /      \  /      \  /
     \/        \/        \/        \/        \/
     /\        /\        /\        /\        /\
    /MM\      /MM\      /MM\      /MM\      /MM\
   / .. \    / .. \    / .. \    / .. \    / .. \
   \ -- /    \ -- /    \ -- /    \ -- /    \ -- /
    \  /      \  /      \  /      \  /      \  /
     \/        \/        \/        \/        \/
```

Awesome ASCII Art: Scary Sharks

Creative Option 2: random length shark



Awesome ASCII Art: Educating Eighths!

```
Creative Option 2: Several eighth-notes in a line, along with some fun music
facts
   ----      ----      ----      ----      ----      ----      ----      ----
  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
(@)) (@)  (@)) (@)  (@)) (@)  (@)) (@)  (@)) (@)  (@)) (@)  (@)) (@)  (@)) (@)

Did you know? There are 2 eighth-notes in a quarter note. You have 8 eight notes,
which equals 4.0 quarter notes!

In 4/4 time, there are 4 quarter notes in a measure. 4.0 quarter notes equals 1.0
full measures!
```

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: we're trying to train our TAs 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!

- "I think that leaving it up to his own judgement is a flawed process. There are things in people's headshots that some people using screen reader might find important that Jake might not."
- "it's never okay to think that people don't "deserve" an alt text"

How could we avoid this? Two main suggestions:

1. by writing alt text for everyone
2. by asking people to write their *own* alt text
   - something for you to consider – e.g., for your club's Instagram!

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)
- on Canvas (Canvas supports alt text for images)
- in design software like PowerPoint (alt text in MS Office), Canva (alt text in Canva), or Figma* (Figma's accessibility mode)
- many others (Webtoons, Discord, Tumblr, Pinterest, new sites, Valorant, broadcast sports, album covers, Reddit, memes!)
- and in different contexts (classes, jobs, RSOs, etc.)

Case Study: Watching Sports

"I watch a lot of sports, which is a good example of constantly moving and changing images. Various camera angles, different views, graphics and stats, and score displays are just some of the components visible on the screen. From my research, I found that alt text is not widely implemented for live broadcasts, because this would require the description to change in real time as the live event unfolds."

Alt text is not *always* right!

- in many cases, concise alt text *is not enough*
  - for information-dense content: video, a poster 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!

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
    **/
}
```

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!
```

B.
```
Welcome!
Hello!
Goodbye!
```

C.
```
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

```
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`

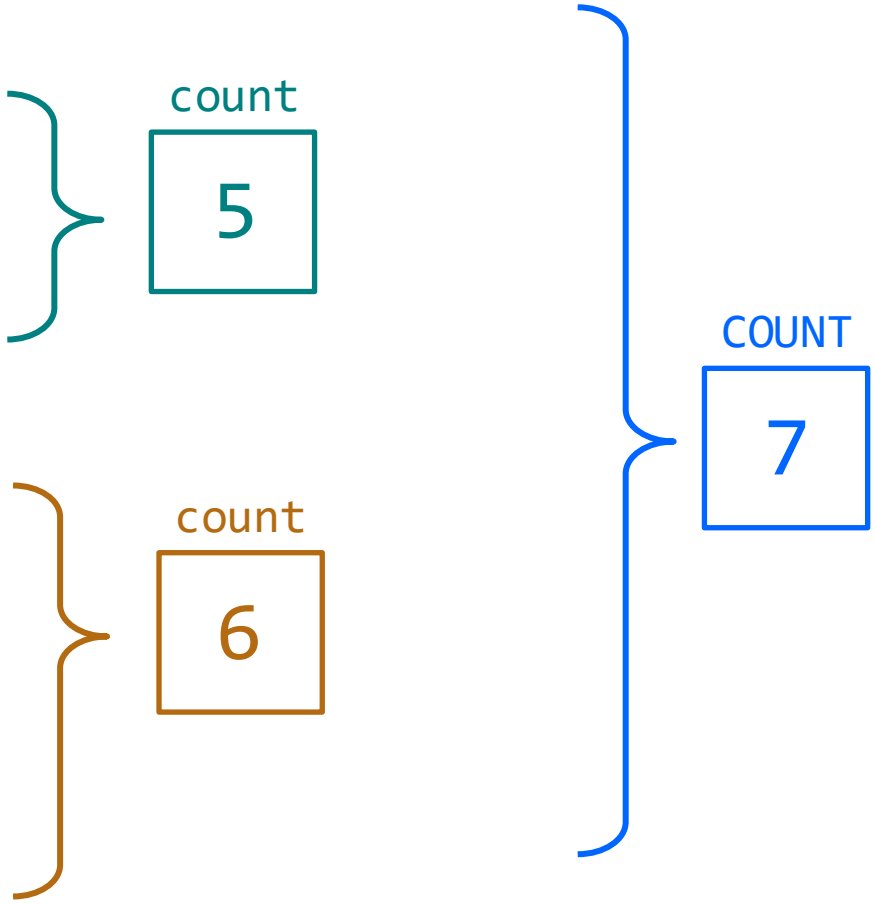
C. `count is: 6`

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();
}
```



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;
    ...
}
```