LEC 07

CSE 121

Methods, Parameters, Returns

Questions during Class?

Raise hand or send here

sli.do #cse121

BEFORE WE START

Talk to your neighbors:

Is cereal a soup?

Music: ❄ CSE 121 26wi Lecture Tunes ❄

Instructor: Miya Natsuhara

TAs:

Amogh	Hayden	Anum	Sam	Shayna
William	Aki	Abdul	Ethan	Jesse
Johnathan	Spencer	Janvi	Jessica	Minh
Anant	Savannah	Navya	Paul	Cayden
Reese	Tamsyn	Ruslana	Carson	

Agenda

- **Announcements, Reminders**
- Methods review
- Parameters review

Announcements, Reminders

- **P1** is out, due next Tuesday, Feb 3rd
  - start early – this one is tough!
- **R1** released yesterday, due Thursday Feb 5th
- **Quiz 0** was yesterday
  - grades will be out before Quiz 1

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!
```

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!");
    ...
}
```

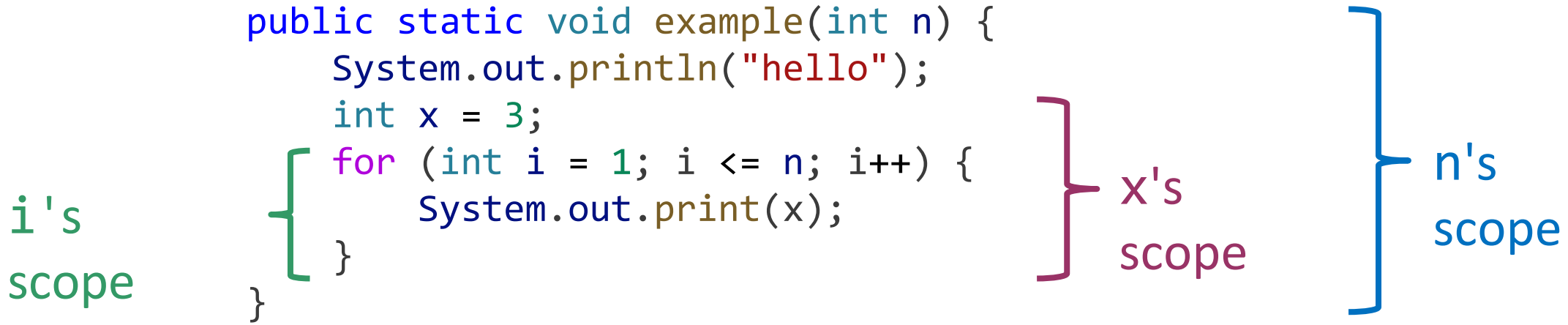
Calling a method with a parameter…

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

PCM: Scope, Redux

Our scope rules also apply to methods and parameters!

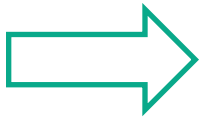
- general rule: from its **declaration to the next closing brace, }**
- a variable declared in a method only exists <u>in that method</u>!



PCM: Stars Example

```
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("*");
    }
}
```

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`

Practice: Pair

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`

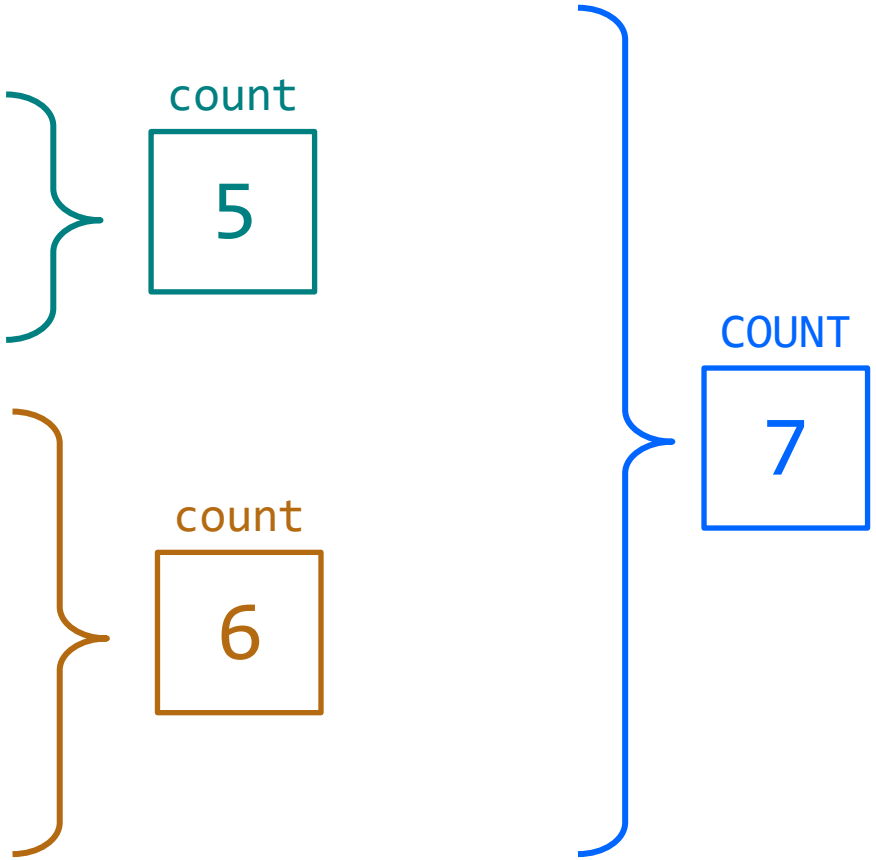
C. `count is: 6`

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();
}
```



New: Method Comments

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

```
// Randomly generates 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 = new Random();
    int num1 = randy.nextInt(10) + 1;
    int num2 = randy.nextInt(10) + 1;
    int sum = num1 + num2;
    ...
}
```