

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 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	

Agenda

- Announcements, Reminders
- Revisit class constants, cumulative sum pattern
- Methods & Parameters Review
- Programming with Methods & Parameters

Announcements, Reminders

- **P1 (Election Simulator)** is out now, due next Tuesday
 - Computer Quiz 1 will be October 28 – November 4
- Remember to email cse121-instructors@cs.washington.edu if you need to reach Miya and me



Practice: Think

sli.do

#cse121

```
String myString = _____;  
for (int i = 0; i < 10; i++) {  
    _____;  
}  
System.out.println(myString);
```

What should fill in each blank so that this prints 0123456789 to the console?

A	B	C	D
"0" myString += "1"	"" myString = i	"0" myString += i	"" myString += i

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!

C.

Welcome! Hello! Goodbye!

B.

Welcome!
Hello!
Goodbye!

D.

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



Practice: Pair



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

```
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

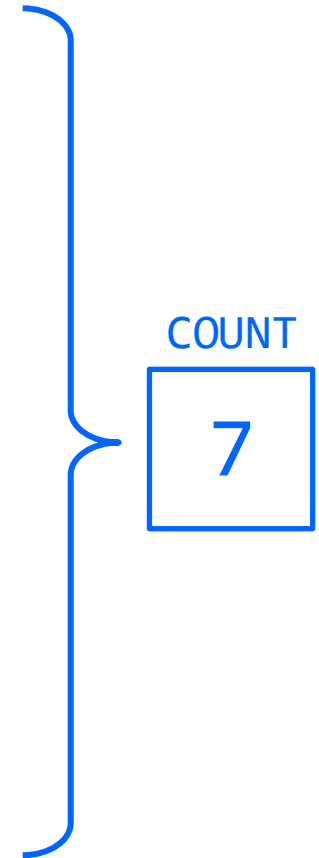
Walkthrough: Counting Counts



```
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!

```
// 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;  
    ...  
}
```