

CSE 121 – Lesson 7

Miya Natsuhara

Spring 2023



Music: [121 23sp Lecture Vibes](#) 

TAs:

<i>Jasmine</i>	<i>Atharva</i>	<i>Mia</i>	<i>Justin</i>
<i>Shananda</i>	<i>Julia</i>	<i>Archit</i>	<i>Aishah</i>
<i>Vidhi</i>	<i>Anju</i>	<i>Grace</i>	<i>Claire</i>
<i>Larry</i>	<i>Lydia</i>	<i>Kailye</i>	<i>Lydia</i>
<i>Jacqueline</i>	<i>Jonus</i>	<i>Joshua</i>	<i>Kai</i>
<i>Afifah</i>	<i>Hugh</i>	<i>James</i>	

[**sli.do #cse121**](#)

Announcements, Reminders

- Programming Assignment 1 is out, due Tues April 25
 - Start early!
 - Math Problems case study video walkthrough posted
- Resubmission Cycle 1 released yesterday, due Thurs April 27
- Quiz 0
 - Plan is to release grades tonight!
 - First opportunity for Quiz 0 Retakes will be Tuesday, April 25
(Retake details and form released tonight)

(PCM) Parameters

A value passed to a method by its caller

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

Calling a method with a parameter...

```
myMethod(42);
```

(Review) Scope

- The part of a program where a variable exists.
 - From its declaration to the end of the { } braces
 - Ex: a variable declared in a for loop only exists in that loop
 - Ex: a variable declared in a method exists only in that method

```
public static void example() {  
    System.out.println("hello");  
    int x = 3;  
    for (int i = 1; i <= 10; i++) {  
        System.out.println(x);  
    }  
}
```

The diagram illustrates the scope of variables `i` and `x` in the provided Java code. A green curly brace on the left, labeled "i's scope", encloses the entire for loop, indicating that `i` is defined within this loop's scope. A red curly brace on the right, labeled "x's scope", encloses the declaration of `x` and the body of the for loop, indicating that `x` is defined within this scope.

Poll in with your answer!



What will be the last line of output after this code has executed?

```
public static void main(String[] args) {  
    int count = 5;  
    line(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();  
}
```

- A. count is: 5
- B. count is: 6
- C. count is: 1
- D. I'm lost

```

public class Scope {
    public static void main(String[] args) {
        int val = 1;
        mOne(val);           // Prints "One: 1"
        val = -1;
        mTwo(val);          // Prints "Two: -2"
        mThree(val);         // Prints "One: -1"
                            // "Three: 2"
    }
}

// Method mOne()
public static void mOne(int val) {
    System.out.println("One: " + val);
}

// Method mTwo()
public static void mTwo(int val) {
    val = val * 2;
    System.out.println("Two: " + val);
}

// Method mThree()
public static void mThree(int val) {
    mOne(val);
    val = val + 3;
    System.out.println("Three: " + val);
}

```

Val ~~1~~ -1

1 -1

-1 -2

1 -1 2

Output:

One: 1

Two: -2

One: -1

Three: 2

Poll in with your answer!



What is the output of this program?

```
public class ParameterMystery {  
    public static void main(String[] args) {  
        int x = 9;  
        int y = 2;  
        int z = 5;  
  
        mystery(z, y, x);  
        mystery(y, x, z);  
    }  
  
    public static void mystery(int x, int z, int y) {  
        System.out.println(z + " and " + (y - x));  
    }  
}
```

Handwritten annotations:

- Variables: x has a value of 9, y has a value of 2, z has a value of 5.
- Call to `mystery(z, y, x);`: z is 5, y is 2, x is 9.
- Call to `mystery(y, x, z);`: z is 5, y is 9, x is 2.
- Final output values: 9 and 3 (highlighted in orange).

- A. 2 and 4
9 and 3

- B. 5 and -7
5 and -7

- C. 9 and -3
5 and -7

- D. I'm lost

Output:

2 and 4

9 and 3