

CSE 121 Lesson 8:

Return Values

Matt Wang
Spring 2024



TAs:	Andy	Anju	Archit	Arkita	Autumn	Christian
	Hannah H	Hannah S	Heather	Hibbah	Janvi	Jessie
	Jonus	Julia	Luke	Maria	Mia	Ritesh
	Shayna	Simon	Trey	Vidhi	Vivian	Gumball?

[sli.do #cse121-8](https://sli.do/#cse121-8)

Today's playlist:
[CSE 121 lecture beats 24sp](#)

Announcements, Reminders

- Creative Project 2 (C2) releasing later today (due Thursday May 2nd)
- R1 due tomorrow; R2 opens tomorrow (due Thursday May 2nd)
 - R2 eligible assignments: C0, P0, C1, P1
- Wednesday May 1st: Mid-quarter Formative Feedback in class
- Quiz 0 is tomorrow in your quiz section!
 - Quiz logistics [announced on Ed](#)
- Quiz review reminders
 - Priority materials: practice quizzes, starred section problems!
 - Ed Class megathreads (great while reviewing lecture)

(Recall) Methods & Parameters

Definition: A value passed to a method by its caller; sending information into a method

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

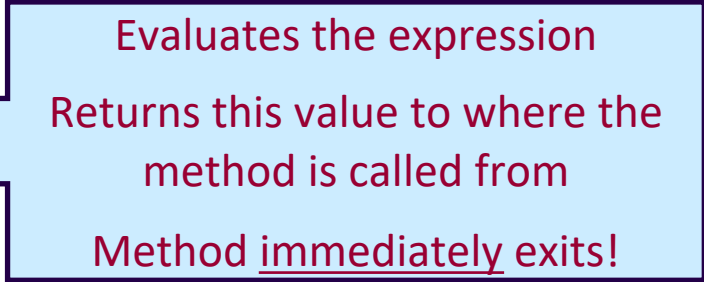
Calling a method with a parameter...

```
myMethod("Laufey"); // Prints out  
                    // "Laufey is the best!"
```

(Recall) Returns 1

Returns allow us to send values out of a method

```
public static <type> myMethod(<zero or more params>) {  
    ...  
    return <value of correct type>  
}
```



Evaluates the expression
Returns this value to where the method is called from
Method immediately exits!

Calling a method that returns a value...

```
<type> result = myMethod(...); // catching what is returned!
```

(Recall) Returns 2

Returns allow us to send values out of a method

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

Calling a method with a parameter...

```
String s = myMethod("Laufey"); // Prints and returns  
                                // "Laufey is the best!"
```

(Recall) Returns 3

Returns allow us to send values out of a method

```
public static String myMethod(String musicalAct) {  
    ...  
    return musicalAct + " is the best!"  
}
```

Calling a method with a parameter...

```
String s = myMethod("Laufey"); // Returns  
                                // "Laufey is the best!"
```

(Recall) String Methods

Usage: `<string variable>.<method>(…)`

Method	Description
<code>length()</code>	Returns the length of the string.
<code>charAt(i)</code>	Returns the character at index <i>i</i> of the string
<code>indexOf(s)</code>	Returns the index of the first occurrence of <i>s</i> in the string; returns -1 if <i>s</i> doesn't appear in the string
<code>substring(i, j)</code> or <code>substring(i)</code>	Returns the characters in this string from <i>i</i> (inclusive) to <i>j</i> (exclusive); if <i>j</i> is omitted, goes until the end of the string
<code>contains(s)</code>	Returns whether or not the string contains <i>s</i>
<code>equals(s)</code>	Returns whether or not the string is equal to <i>s</i> (case-sensitive)
<code>equalsIgnoreCase(s)</code>	Returns whether or not the string is equal to <i>s</i> ignoring case
<code>toUpperCase()</code>	Returns an uppercase version of the string
<code>toLowerCase()</code>	Returns a lowercase version of the string

String example

```
String s = "bubblegum";
```

```
s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";
```

```
s = "g".toUpperCase() + s.substring(8) + "ball";
```

```
s = "G" + s.substring(8) + "ball";
```

```
s = "G" + "um" + "ball";
```


Example of returns: Math class

Methods	Returns
<code>Math.abs(<i>value</i>)</code>	Absolute value of <i>value</i>
<code>Math.ceil(<i>value</i>)</code>	<i>value</i> rounded up
<code>Math.floor(<i>value</i>)</code>	<i>value</i> rounded down
<code>Math.max(<i>value1</i>, <i>value2</i>)</code>	Larger of the two given values
<code>Math.min(<i>value1</i>, <i>value2</i>)</code>	Smaller of the two given values
<code>Math.round(<i>value</i>)</code>	<i>value</i> rounded to the nearest whole number
<code>Math.sqrt(<i>value</i>)</code>	Square root of <i>value</i>
<code>Math.pow(<i>base</i>, <i>exp</i>)</code>	<i>base</i> to the <i>exp</i> power

Math example

```
double value = 823.577564893;  
double roundedValue = (double) Math.round(value * 100) / 100;  
                        = (double) Math.round( 82357.7564893 ) / 100;  
                        = (double) 82358.0 / 100;  
                        = 823.58
```

Poll in with your answer!

To go from Celsius to Fahrenheit, you multiply by 1.8 and then add 32.
Which of these correctly implements this logic as a method?



[sli.do #cse121-8](https://sli.do/#cse121-8)

A.

```
public static void celsiusToF(double celsius) {  
    double fahrenheit = celsius * 1.8 + 32;  
    return fahrenheit;  
}
```

B.

```
public static void celsiusToF(double celsius) {  
    double fahrenheit = celsius * 1.8 + 32;  
}
```

C.

```
public static double celsiusToF(double celsius) {  
    int fahrenheit = celsius * 1.8 + 32;  
    return fahrenheit;  
}
```

D.

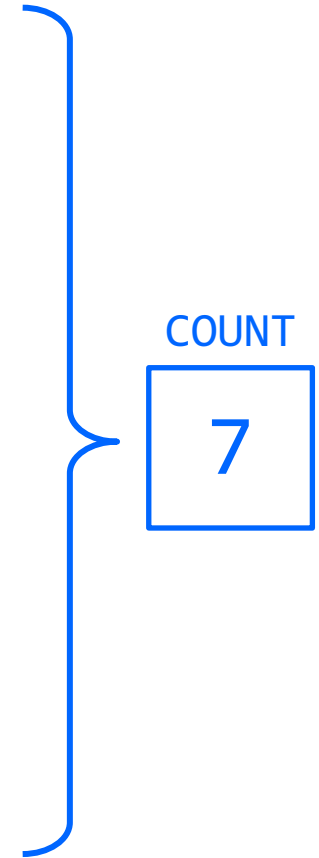
```
public static double celsiusToF(double celsius) {  
    return celsius * 1.8 + 32;  
}
```

(Recall) Tricky Poll: Last line printed?

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



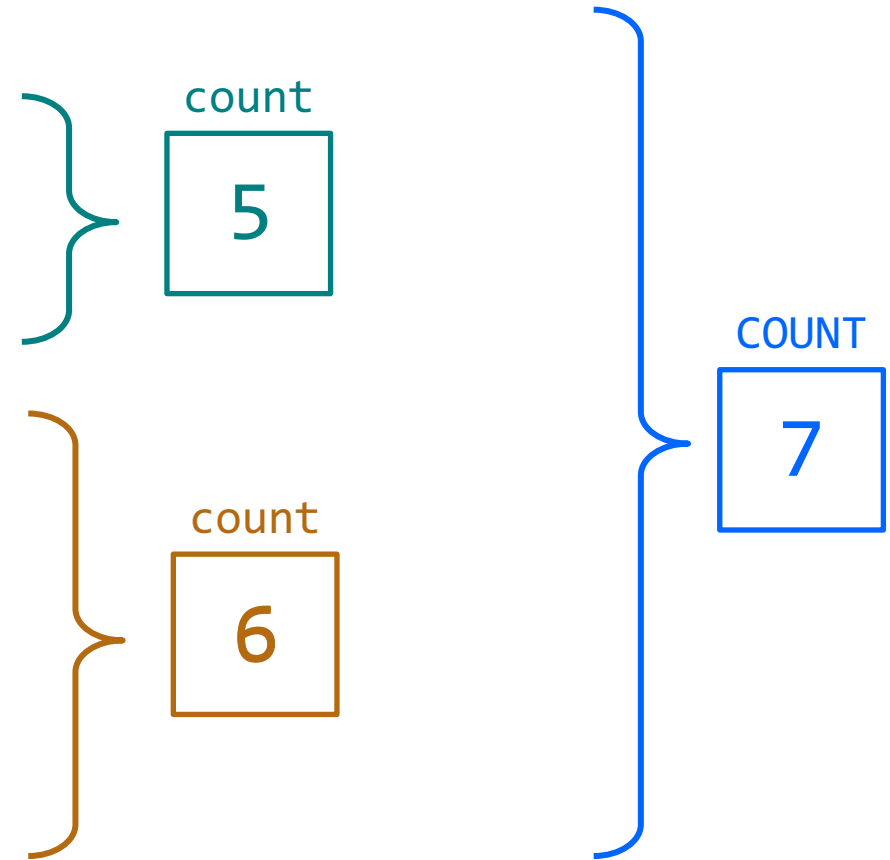
```
public static void line(int count) {  
    for (int i = 1; i <= count; i++) {  
        System.out.print("*");  
    }  
    count++;  
    System.out.println();  
}
```



Tricky Poll: Returnable

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

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



Poll in with your answer!



[sli.do #cse121-8](https://sli.do/#cse121-8)

What value is returned from this method?

```
public static int returnExample() {  
    for (int i = 0; i < 5; i++) {  
        return i;  
    }  
    return -1;  
}
```

A. -1

B. 0

C. 4

D. 5

Common Problem-Solving Strategies

- **Analogy** – Is this similar to another problem you've seen?
- **Brainstorming** – Consider steps to solve problem before jumping into code
 - Try to do an example "by hand" → outline steps
- **Solve sub-problems** – Is there a smaller part of the problem to solve?
- **Debugging** – Does your solution behave correctly?
 - What is it doing?
 - What do you expect it to do?
 - What area of your code controls that part of the output?
- **Iterative Development** – Can we start by solving a different problem that is easier?