

# CSE 121 – Lesson 8

Miya Natsuhara

Autumn 2023

Music: [121 23au Lecture Tunes](#) 🐼

*bonus!*  
*(open story)*



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

TAs:	Trey	Christina	Sahej	Vinay	Kriti
	Sebastian	Colton	Anju	Maria	Minh
	Annie	Janvi	Jonus	Shreya	Vivian
	Jasmine	Arkita	Lydia	Andy	Nicole
	Christian	Vidhi	Luke	Nicolas	Simon
	Lucas	Ritesh	Andras	Shayna	Jessie
	Logan	Hibbah	Archit	Hannah	Lydia
	Jacob	Julia	Ayesha	Aishah	Yijia



# Announcements, Reminders

- Programming Assignment 1 was due last night
- Creative Project 2 will be released later today
- Quiz 1 scheduled for Thursday next week, Nov 2
- Wednesday Nov 1: Mid-term Formative Feedback *in class*

# 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?

# Metacognition

**Metacognition:** thinking about how you think

Asking questions about your solution process

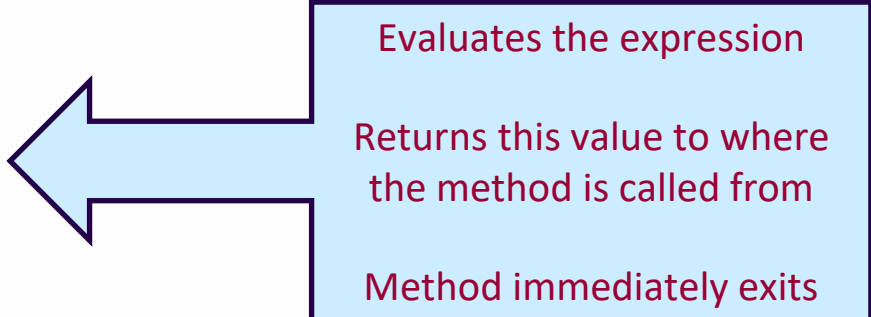
Examples

- **While debugging:** explain to yourself why you're making this change to your program
- **Before running your program:** make an explicit prediction of what you expect to see
- **When coding:** be aware of when you're not making progress, so you can take a break or try a different strategy
- **When studying:** What is the relationship of this topic to other ideas in the course?

# (PCM) Returns

Returns allow us to send values *out of a method*

```
public static <type> myMethod(int num) {  
    System.out.print(num + " is the best!");  
    ...  
    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(42);
```

# (Recall) String Methods

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

Method	Description
<code>length()</code>	<b>Returns</b> the length of the string.
<code>charAt(i)</code>	<b>Returns</b> the character at index <i>i</i> of the string
<code>indexOf(s)</code>	<b>Returns</b> 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>	<b>Returns</b> 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>	<b>Returns</b> whether or not the string contains <i>s</i>
<code>equals(s)</code>	<b>Returns</b> whether or not the string is equal to <i>s</i> (case-sensitive)
<code>equalsIgnoreCase(s)</code>	<b>Returns</b> whether or not the string is equal to <i>s</i> ignoring case
<code>toUpperCase()</code>	<b>Returns</b> an uppercase version of the string
<code>toLowerCase()</code>	<b>Returns</b> a lowercase version of the string

# String example

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

# Poll in with your answer!



What is the correct implementation of a `maxDatingAge` method?

**A.**

```
public static void maxDatingAge(int age) {  
    int maxDatingAge = age - 7 * 2;  
    return maxDatingAge;  
}
```

**B.**

```
public static void maxDatingAge(int age) {  
    int maxDatingAge = age - 7 * 2;  
}
```

**C.**

```
public static int maxDatingAge(int age) {  
    int maxDatingAge = (age - 7) * 2;  
    return maxDatingAge;  
}
```

**D.**

```
public static int maxDatingAge(int age) {  
    return (age - 7) * 2;  
}
```

# Poll in with your answer!



What value is returned from this method?

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

A. 0

B. 4

C. 5

D. -1