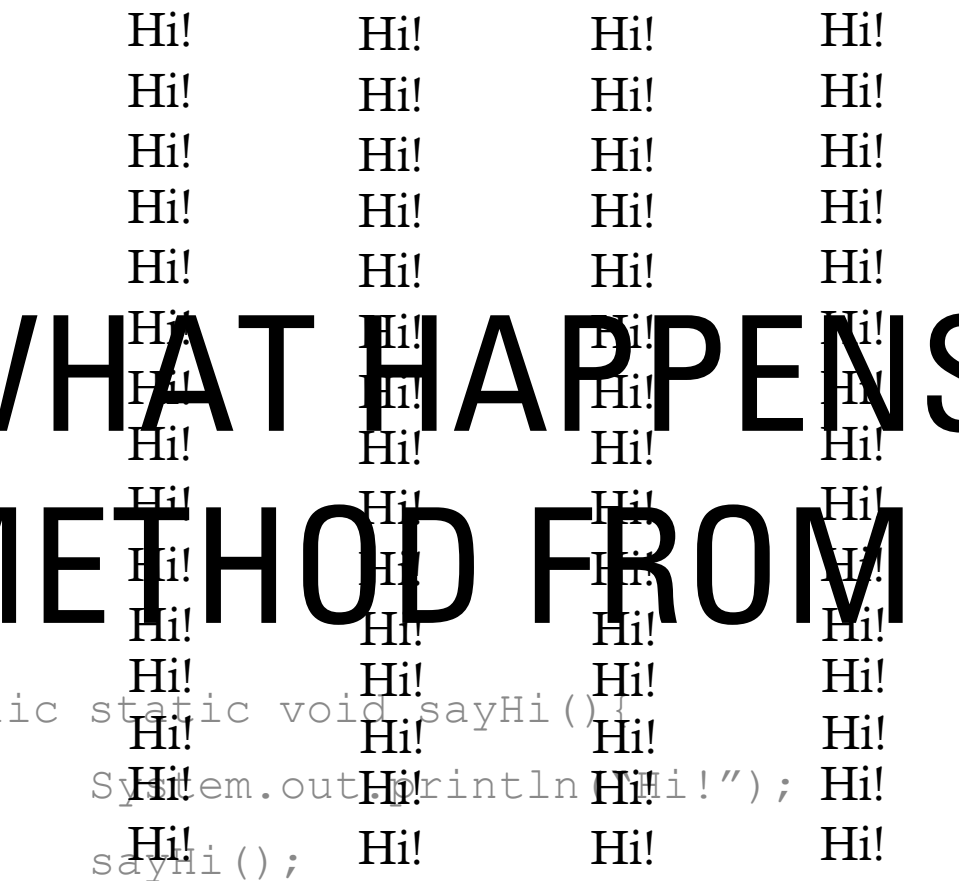


---

# WEEK TEN

Acknowledgements: Slides created based off material provided by Dr. Michael Raymer and Dr. Travis Doom

---



WHAT HAPPENS IF I  
CALL A METHOD FROM INSIDE  
A CLASS?

```
public static void sayHi() {
    System.out.println("Hi!");
    sayHi();
}
```

Excuse me!

Exception in thread "main" java.lang.StackOverflowError

---

# WHAT JUST HAPPENED?

Recursion!

---

# RECURSION

- **Recursive method:** any method that calls itself
- Starts with a complicated problem and breaks it down into smaller and smaller problems
  - aka divide and conquer decomposition
- Why use recursion?
  - Sometimes simpler/easier to understand and maintain
  - Sometimes more flexible
  - Sometimes faster
  - **Short answer:** it's another tool in your toolbox

---

# OKAY, BUT HOW DO WE AVOID A STACK OVERFLOW ERROR?

An endpoint

Termination condition

A base case!

$$\lambda_{\text{um}} = 147$$

- process t  
47  
147

4

$$4 + 7 = 11$$

1

+

11

3/31/2025

6

12

---

# HOW TO UNDERSTAND RECURSION

- A few very simple base cases (often just one)
- Rules to break down complex cases into simpler cases of the same general problem
- Are we done yet? If so, return the results.
- If not, simplify the problem (move towards the base case), by constructing a solution from smaller similar problems
- If you have an already solved similar but simpler problem, how can that help you solve a more complex problem?

---

# PRACTICE TOGETHER

- Reverse a word
- **Input:** hello    **output:** olleh
- What is our base case?
- What small step can we take before making the recursive call?



---

# YOUR TURN

- Write a recursive method that counts how many even numbers are in a String
- You can assume the String will only include numbers
- **Input:** “2395832”      **Output:** 3

---

# HOW ABOUT ONE MORE?

- Write a recursive method that takes in an ArrayList of comparable generic types
- The method should find and return the “max” value of the ArrayList
- **Input:** [4, 1, 4, 6, 3, 9, -3, -6, 8]                      **Output:** 9
- **Input:** ["hi", "smile", "zoo", "laugh", "kind", "apple"]                      **Output:** “zoo”

---

# LET'S TRY SOMETHING A BIT MORE DIFFICULT

- Write a recursive method that takes in a single word and returns every possible permutation of that word in a list
- **Input:** "cat"    **Output:** [cat, cta, act, atc, tca, tac]
- What's our base case going to be?
- How do we break the problem down?
- What other structures may we need to use?

---

# WHAT ACTUALLY HAPPENS WHEN A RECURSIVE CALL OCCURS IN A LOOP?

