

LEC 09

**CSE 123**

# Recursive Programming

Questions during Class?

Raise hand or send here


sli.do #cse123



BEFORE WE START

*Talk to your neighbors:**What are your spring break plans?***Instructors:** Brett Wortzman  
Miya Natsuhara**TAs:**Music: CSE 123 25wi Lecture Tunes

# Announcements

- Resubmission Period 2 due tonight (2/7) at 11:59pm
  - Last opportunity for C0
- Quiz 1 Tuesday (2/11) in your registered section
- Programming Assignment 1 is due Wednesday (2/12) at 11:59pm
- Resubmission Period 3 opening tonight, due next Friday (2/14 )
  - Assignments available: P0, C1

# Recursive Methods [Review]

- 2 components of every recursive method:
- Recursive case
  - Decompose problem into subproblem
  - Make the actual recursive call
  - Combine results meaningfully
- Base case
  - Simplest version of the problem
  - No subproblems to break into
  - Return known answer



*If decomposing moves you closer to the base, no infinite recursion!*

# Why Recursion?

- Generally, anything you can write iteratively you can write recursively
  - So why write anything recursively?

*Recursion is particularly useful when dealing with something that's recursively defined*

- Math examples:
  - Factorial:  $n! = n * (n - 1)!$
  - Exponent:  $x^n = x * x^{n-1}$
  - Fibonacci:  $fib(n) = fib(n - 1) + fib(n - 2)$
- Non-math examples?
  - ListNodes (int data, ListNode next)
  - Other ideas?

# Files

- We'll say that computer files fall into one of the following categories:



Standard file (.txt, .csv, .java)

```
f.isDirectory() -> false
```



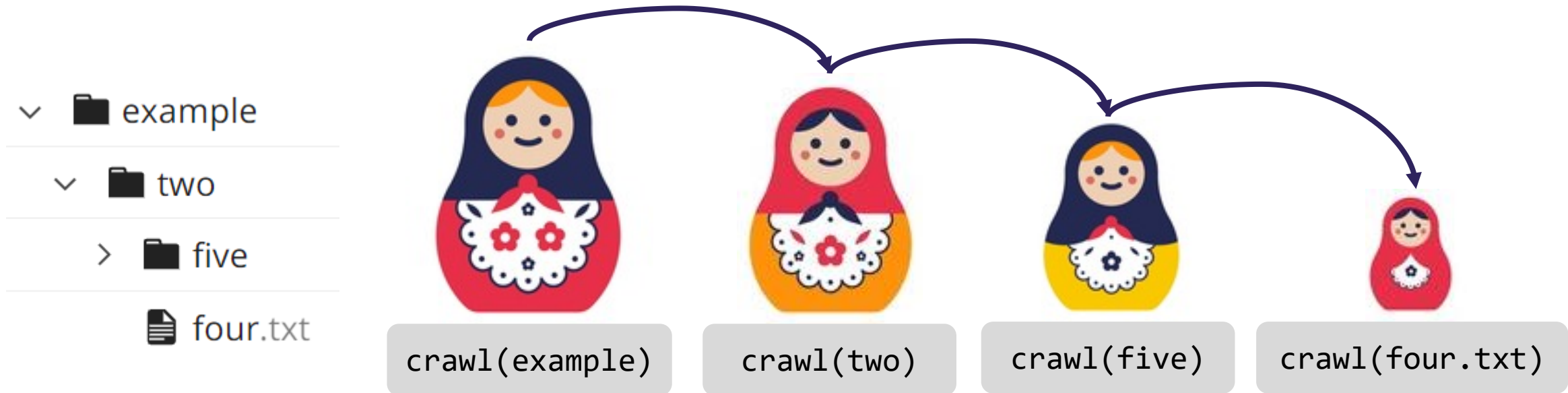
Directory w/ subfiles

```
f.isDirectory() -> true  
File[] subFiles = f.listFiles()
```

*This is a recursive definition! A File is either normal, or a directory with a File[] of subFiles*

# Crawl w/ Indentation

- How can one of our files know what level it's on?



- What if a bigger doll told the next smaller doll the level?
  - So long as the first doll is told the right value, this will work!
- Remember, recursive method calls are still method calls
  - How can we pass information from a bigger doll to a smaller doll?

# Public / Private Pairs

- Used when we need additional information between recursive calls
- Private helper method hides additional info
  - Clients shouldn't have to worry about it
- All public method does is kick-start the private one
  - What's the correct starting value(s) for additional param(s)?

*Question to ask: "Do I need to keep track of any additional information?"*