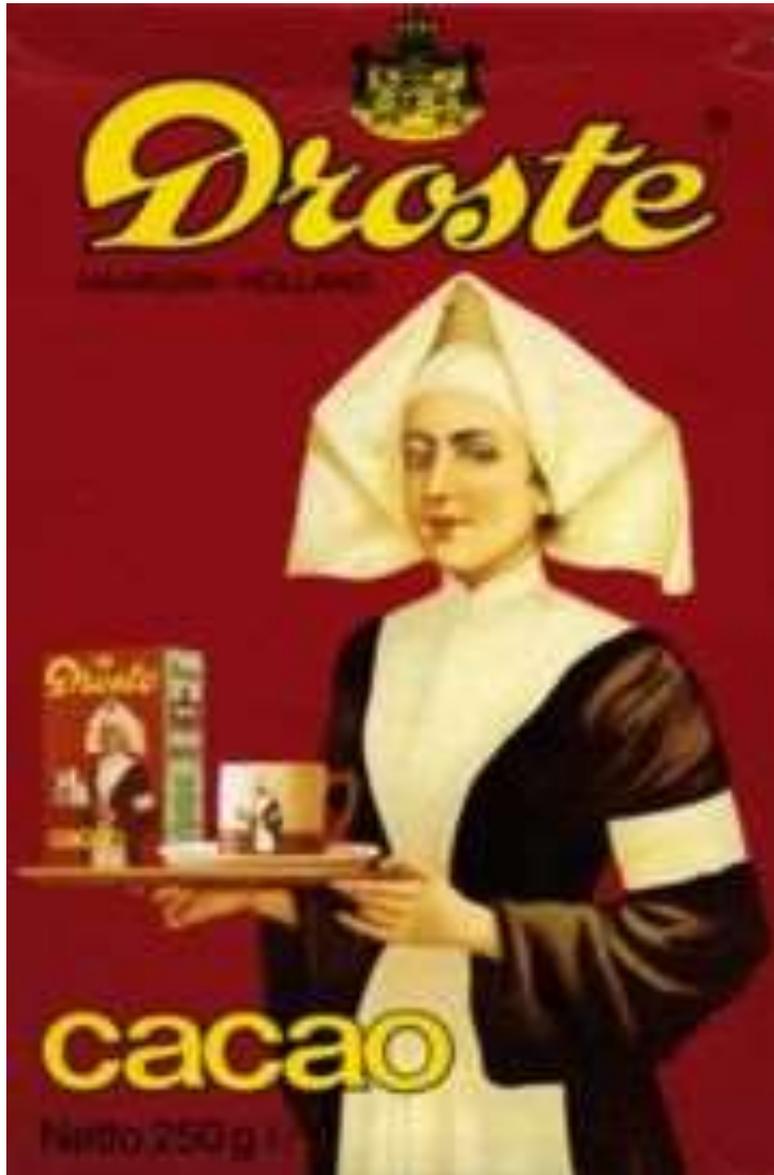


# Review and Introduction to Recursion

- Turn in your Design Problem 3 solution



# CSSE 220: New Material

Recursion

Import *Recursion* project from the repo

# Announcements

- The next 4 class days:
  - A new way to think: **Recursion**
  - A new way to break up and re-use code:  
**Interfaces**
    - Making interactive apps requires this

# Recursion

- A solution technique where the same computation **occurs repeatedly** as the problem is solved



recurs

- Examples:

- Sierpinski Triangle:

- [https://en.wikipedia.org/wiki/Sierpinski\\_triangle](https://en.wikipedia.org/wiki/Sierpinski_triangle)

- Towers of Hanoi:

- <http://www.mathsisfun.com/games/towerofhanoi.html>

- <https://www.cs.cmu.edu/~cburch/survey/recurse/hanoiex.html>

- or search for Towers of Hanoi

# An example – Triangle Numbers

- If each red block has area 1, what is the **area  $A(n)$**  of the Triangle whose *width* is  $n$ ?

– Answer:

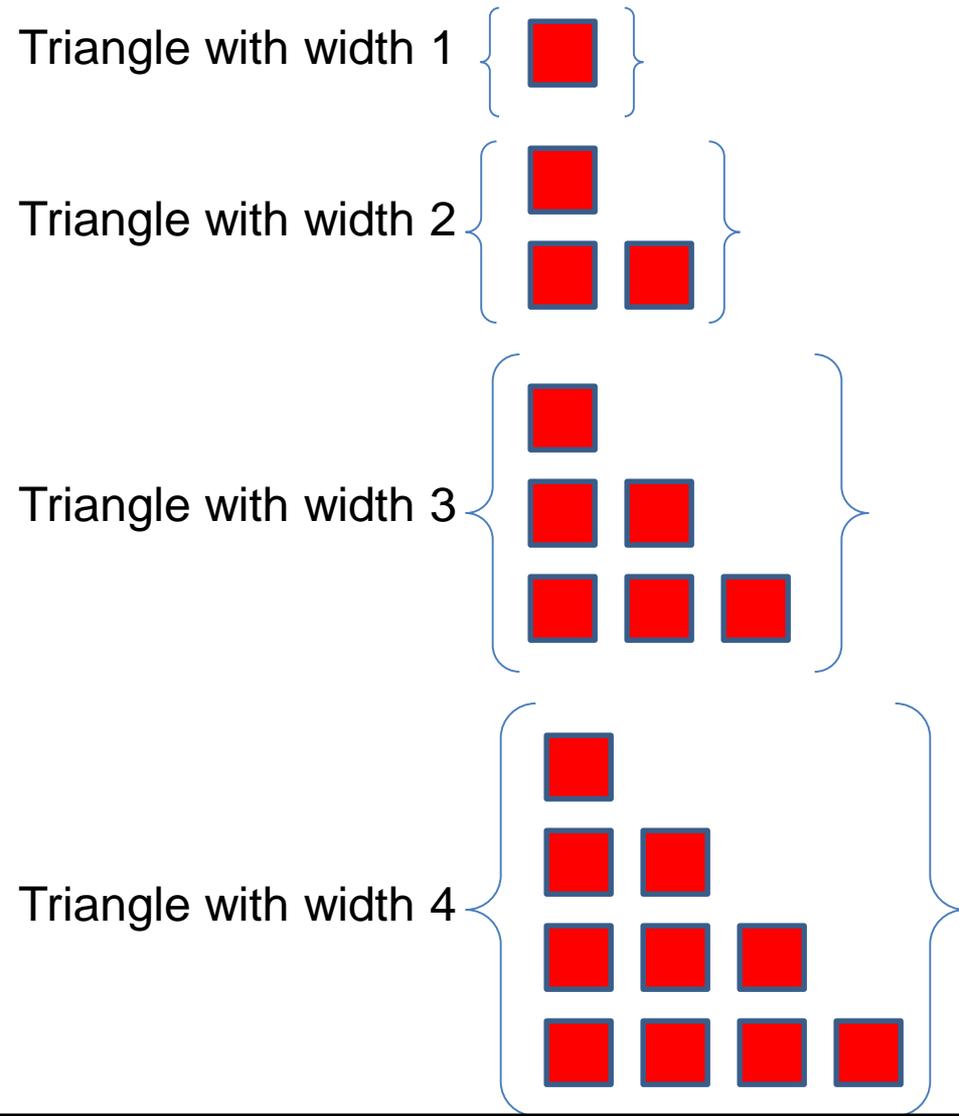
$$A(n) = n + A(n-1)$$

- The above holds for which  $n$ ? What is the answer for other  $n$ ?

– Answer: The recursive equation holds for

$$n \geq 1.$$

For  $n = 0$ , the area is 0.



# Key Rules to Using Recursion

- ▶ Always have a **base case** that **doesn't recur**
- ▶ Make sure recursive case always **makes progress**, by **solving a smaller problem**
- ▶ **You gotta believe**
  - Trust in the recursive solution
  - Just consider one step at a time

# Frames for Tracing Recursive Code

1. Draw box when method starts

2. Fill in name

method name (args)

parameters and local variables

base case condition(s)

return statement

3. List every parameter and its argument value.

5. Check Condition(s)

4. List every local variable declared in the method, **but no values yet**

6. Add box for next recursive call frame. Add blank for unknown value

7. Add blank for unknown value, if needed (may be box from #6)

8. Step through the method, update variable values, draw new frame for new calls

Thanks to David Gries for this technique

Q1-Q3

# Programming Problems

- Add recursive isPalindrome methods to SimplePalindrome
- Optionally add a recursive method to Sentence for computing whether Sentence is a palindrome

Sentence
String text
String toString() boolean isPalindrome()

# Practice Practice Practice

- Head to <http://codingbat.com/java/Recursion-1> and solve 5 problems. I personally like bunnyEars, bunnyEars2, count7, fibonacci, and noX
- Get help from me if you get stuck
- Then take a look at the recursion homework