

# CSSE 220 Day 20 Recursion

Checkout Recursion project from SVN

#### Questions?

# Gödel, Escher, Bach

- By Douglas Hofstadter
- Argues that intelligence arises (in part) because of our ability to think about thinking



#### Recursion

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



#### 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 what n? What is the answer for other n?
  - Answer: The recursive equation holds for n > 1.

For n = 1, the area is 1.

Let's see how this translates naturally to code. Then let's trace the execution of the code (next slide).





# **Tabletop Roleplaying**



#### **Programming** Problem

- Add a recursive method to Sentence for computing whether Sentence is a palindrome
  - A *palindrome* is a String that is

Examples of palindromes from http://www.fun-withwords.com/palin\_example.html Never odd or even Murder for a jar of red rum May a moody baby doom a yam? Go hang a salami; I'm a lasagna hog! **Oozy rat in a sanitary zoo** Do geese see God?

the same backwards as forwards	
the same backwards as forwards	Sentence
<ul> <li>We will ignore punctuation, spaces, and case.</li> </ul>	
• Key idea: use the definition of <i>isPalindrome</i> ()	String text
make progress to a smaller problem?	String toString()
• Here.	boolean equals()
x.isPalindrome() iff	boolean isPalindrome
isPalindrome()?	on't worry about punctuation, spaces Id case at this point of your thinking.
<pre>• x.isPalindrome() iff</pre>	
<pre>xMinusFirstAndLastLetter.isPalindrome()</pre>	and?
.isPalindrome() iff xMinusFirstAndLastLetter and first letter equals last letter	<mark>c.isPalindrome()</mark> Q10

#### **Recursive Functions**



### **Recursive Helpers**

- Our isPalindrome() makes lots of new Sentence objects
- We can make it better with a "recursive helper method"
  - Many recursive problems require a helper method



## **Key Rules to Using Recursion**

- Always have a base case that doesn't recurse
- 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

# **Another Definition of Recursion**

If you already know what recursion is, just remember the answer. Otherwise, find someone who is standing closer to Douglas Hofstadter than you are; then ask him or her what recursion is."

—Andrew Plotkin

#### Sierpinski

// Draws the first equilateral triangle // called for by the algorithm. Point2D p1 = new Point2D.Double( left, bottom); Point2D p2 = new Point2D.Double( left + base, bottom); Point2D p3 = new Point2D.Double( left + base / 2.0, bottom - base \* HEIGHT TO BASE RATIO); Shape triangle = makeTriangle( p1, p2, p3); q.setColor(Color.RED); g.fill(triangle);

TODO Implement rest of this method.



Exam 2 is next Thursday evening. Major topics are:

- UML class diagrams
   and how to implement them
- event-driven programming
- GUI programming
- polymorphism
- interfaces
- inheritance
- recursion

# Work Time

# Work on VectorGraphics with your team

- Cycle 1 code deliverables are due Monday.
- Cycle 2 planning deliverables are due Tuesday.
   Sierpinski is due Tuesday (not Monday)