Program 3 assigned – FINISH IT BEFORE BREAK

Last Time
PriorityQueue Review
Java’s Comparable Interface
Heap Data Structure
• insert
• removeMax
Java’s Stack, Queues, PriorityQueues
Call Stack Tracing

Today
Call Stack Tracing (from last time)
Recursion
• recursion vs. iteration
• rules of recursion
• constructing recursive code
• practice writing recursive code
Exam 1 returned

Next Time
Read: continue Recursion
Recursion
★ more practice writing recursive code
★ complexity of recursive methods
★ practice analyzing complexity
Recursion vs. Iteration

Recursion is like iteration:

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Recursion</th>
</tr>
</thead>
</table>

Recursion is NOT like iteration:

- Each loop iteration
- A loop with a bad stopping condition

Rules for Recursion

1.

2.
Recursion

What is it?

Why use it?

→ How would you modify the print method to display a singly-linked chain of nodes in reverse order?
Factorials: $n!$

Consider the factorial of $n$ (assume $n \geq 0$):

$$n! = n \times (n-1) \times (n-2) \times (n-3) \times \ldots \times 2 \times 1$$

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$$

Method Call:

```java
    factorial(6);
```

Iterative Implementation:

```java
    int factorial(int n) {
        int result = 1;
        for (; n > 1; n--)
            result = result * n;
        return result;
    }
```

Recursive Definition:

→ Complete the Recursive Implementation:

```java
    int factorial(int n) {
```
Constructing Recursive Code

→ Write a recursive method that computes \( n^m \)
   that is, it computes double \( n \) raised to an int power \( m \)?

recursive definition:

recursive implementation:

Key Questions:

1.

2.

3.

4.
Practice – ListADT

→ Write a recursive method that displays the values in a (non-null) list of strings.

1.

2.

3.

4.

void display(ListADT<String> list) {