**Recursion**

Recursion is characterized by defining an entity in terms of itself.

**Recursion and the java stack**

A typical recursive method:

int foo(int n) {

int answer = 1;

if (n > 0){

answer = 2 \* foo(n-1);

}

return answer;

}

The JVM creates a new stack frame for each method call.

In order to calculate foo(20), at least 20 stack frames will get pushed to and popper from the Thread’s Java Stack. Very time consuming.

**Many examples in lecture**

**Summary**

Recursive programming is a powerful problem solving technique which can lead to concise, elegant programs.

In some cases, the recursive solution can be terribly inefficient though so it must be used with care or an iterative solution found instead.

Tail recursion is a particular form of recursive call whose execution can be optimized by certain compilers (not Java though).

A recursive method is tail recursive when recursive call is the last thing executed by the function.