

## Recursion

Recursion is using a method or class within itself to repeatedly call itself to reduce the function being performed to a basic state. The reduced state that returns the final output is referred to as the base case. The case that calls the method again is the recursive case. Each time the method calls itself it reduces the input to get closer to the base case. In theory a recursive method can have any number of base cases or recursive cases but most will only have one or two. When using recursion you must be careful to not create an infinite loop. A slightly different method of recursion is tail recursion. This is adding a call to a method at the end of itself to make it repeat over and over. A stop case is needed whenever tail recursion is done to prevent infinite loops. Recursion is useful in performing complex mathematical operations and creating lists of numbers where the next term depends on the previous terms such as the Fibonacci sequence. Recursion can also be performed by creating two methods that call each other. Other functions that use recursion are factorials and solving the towers of Hanoi game.

Example of recursion that gives the factorial of a number:

```
public int factorial(int n){
    if(n < 0){
        return -1;
    }
    else if (n == 0){
        return 1;
    }
    else{
        return n*factorial(n-1);
    }
}
```

Sources: StackOverflow.com