## Lake Washington Institute of Technology

## CS 143 Computer Science II

## Programming Assignment 4

## Recursion and Exception Handling

### Recursive Sum of Numbers

Write a recursive public static method named recursiveSum that accepts a single positive integer parameter and returns the sum of all t he integers from 1 up to the number passed as an argument. For example, if 5 is passed as an argument, the method will return the sum:

1+2+3+4+5

If an argument less than 1 is passed to the method, throw IllegalArgumentException.

#### Requirements

Your implementation of recursiveSum must not use any static fields. Only local variables and parameters can be used.

Test this method from method main using hardcoded values to the call to recursiveSum. Include a test with an illegal argument (with a try/catch structure to demonstrate the exception handling).

Add this method to a project and class named RecursionAndExceptions.

### Recursive Sum of Array Elements

Write a public static method named arraySum that accepts an int array as a parameter and returns the sum of all the elements in the array. This public method should call a private method named arraySum that returns an int, and takes two parameters:

* An array of ints
* An index of the current element in the array

The private version of arraySum must call itself recursively summing another element of the int array parameter with each call.

#### Requirements

Your implementation of arraySum must not use any static fields. Only local variables and parameters can be used.

Test this method with calls from method main using hardcoded values to the public version of the method arraySum. Include a test with an illegal argument (with a try/catch structure to demonstrate the exception handling).

Add these methods to a project and class named RecursionAndExceptions.

### Recursive Print

Implement a method named printPattern that accepts an integer parameter, returns void, and outputs a number pyramid (see below). The method must throw an IllegalArgumentException if an argument is passed to the method that is less than or equal to 0.

#### Requirements

There must be no loop structures used in your implementation. All repetitions must be implemented with recursion. This means you will need to write additional recursive methods as well as non-recursive methods to replace nested loops the nested loops that would otherwise be needed.

Try having no more than 2 recursive methods in your implementation. The number of non-recursive methods is unlimited.

Your methods must not use any static fields. Only local variables can be used in the method implementations.

Test this method with calls from method main using hardcoded values to the public version of the method printPattern. Include a test with an illegal argument (with a try/catch structure to demonstrate the exception handling).

Add the methods to a project and class named RecursionAndExceptions.

#### Print Pyramid

If printPattern is passed an argument of 5, it should output the following:

1

212

32123

4321234

543212345