**INFSCI 0017 – Fundamentals of Object-Oriented Programming (Fall 2018)**

**Lab 4**

## Topics Reviewed

1. Control flow statements
   1. If / else
   2. Loops (for, while)

## Grading and Submission

You are to write a complete Java program that meets the requirements outlined in the Lab 2 Tasks section.

Once you have completed the program, you should demonstrate your program for your Lab TA.

There will be 5 points for this lab, broken down in the following way:

1. Program correctly receives and converts user input (1 point)
2. Program calculates the logarithm correctly (2 points)
3. Program handles negative X by requiring re-entry (2 point)

Note that if your program does not compile, the TA will not grade it.

## Background

A logarithm is a very useful mathematical function. Using a given base, b, we define Y = logbX to be an exponent such that bY = X. If we are dealing exclusively with integer values, we typically round the log up or down, using the ceiling or floor function. Consider rounding down, using the floor function. This will give us the exponent Y such that bY <= X but bY+1 > X.

For example, consider b = 10 (i.e. we are considering base 10 logarithms):

floor (log10 (123456)) = 5 because 105 = 100000 <= 123456 and 106 = 1000000 > 123456

As another example, consider b = 2 (i.e. we are considering base 2 logarithms):

floor (log2 (64)) = 6 because 26 = 64 <= 64 and 27 = 128 > 64

One way of calculating the floor of the logb X is by repeated integer division. Divide by the base until the quotient is less than the base. The number of completed divisions is equal to the floor of the logbX .

For example, consider the cases above, and one additional case:

**Example 1 ⇒ 5 divisions**

1. 123456 / 10 = 12345
2. 12345 / 10 = 1234
3. 1234 / 10 = 123
4. 123 / 10 = 12
5. 12 / 10 = 1 < 10

**Example 2 ⇒ 6 divisions**

1. 64 / 2 = 32
2. 32 / 2 = 16
3. 16 / 2 = 8
4. 8 / 2 = 4
5. 4 / 2 = 2
6. 2 / 2 = 1 < 2

**Example 3 ⇒ 3 divisions**

1. 4567 / 10 = 456
2. 456 / 10 = 45
3. 45 / 10 = 4 < 10

## Lab 4 Tasks

* Ask the user to enter 2 values: the value for which you want to calculate the *log* and the *base*
* The program must calculate a logarithm using the value and the base provided by the user
* The program must handle negative X by requiring re-entry (1 point)