

## **COE318 – Lab 3: Linked Counters**

### **Objectives**

- Implement a Counter class.
- Learn how objects can be linked together.
- Use an “if” statement.

**Duration:** one week.

### **Grading Scheme:**

50% submitted source code

25% in-class demonstration and questions (during week 5 lab hours)

25% in-class quiz – Held during the first 5 mins of the lab class (during week 5 lab hours)

### **Overview**

In mathematics, a number is expressed in positional notation to a certain base, B

$$d_n d_{n-1} \dots d_1 d_{0_B} = \sum d_i B^i$$

For example, the 3-digit number 123 in base 4 represents  $16+8+3=27$  (base 10).

In this lab each digit is represented as a Counter object. A Counter object has an optional left neighbour which is also a Counter object. (The absence of a left neighbour is indicated with the keyword `null`).

The important methods to implement are `getCount()` and `increment()`. If there is no left neighbour, the count is the same as the digit.

If there is a left neighbour, the count is the sum of the digit and the modulus times the count of the left neighbour.

The `increment()` method increments the Counter's digit and, if it reaches its maximum (modulus) value, it is reset to zero. Furthermore, if there is a left neighbour and if the Counter has rolled over, its left neighbour should be incremented as well.

### **Source Code**

Similar to the previous labs, `Counter.java` and `CounterTry.java` are provided with the handout.

### **Step 1: Create a Netbeans Project**

1. Create a Netbeans project called `Counter` which should be placed in a folder called `lab3` (all lowercase and no spaces). The `lab3` folder should itself be in your `coe318`

folder.

2. Create a Java file (class library type) called `Counter`; set the package to `coe318.lab3`; then copy and paste the provided source code.
3. Similarly, create the Java file `CounterTry`. (Ensure you use the same `coe318.lab3` package name.
4. Generate the javadocs and compile and run the project.
5. It should compile correctly and produce output. Unfortunately, the output is incorrect and you have to fix it.

## **Step 2: Add instance variables and fix constructor and getters**

1. Add instance variables for the two components of a counter.
2. Modify the constructor so that they are properly initialized.
3. Fix the remaining methods so that they work for a simple counter without a left neighbour.

## **Step 3: Fix remaining methods**

1. Fix the remaining methods.

## **Step 4: Submit your lab**

You must submit your lab electronically on D2L. Please make sure you hand over the quiz answer sheet to the TA at the end of the in-class quiz.

Please zip up your NetBeans project containing all source files and submit to the respective assignment folder on D2L.