

## COMPX201 & COMPX241 Assignment Two

### Stack

**Due:** Friday 29th April 11:59pm.

#### Part One:

In this assignment you will write a program which processes commands in LIFO order. You will need to define Java classes to implement a stack. Your stack should use a recursive dynamic linked list (i.e. not an array) following the specification given below.

1. **Stack:** define a class called `Stack` in a file called `Stack.java`. This class is to implement a dynamic linked list as a stack that supports the following public methods using recursion where applicable.
  - `push(String x)` - add item `x` to the front of the list.
  - `pop()` - remove an item from the front of the list.
  - `peek()` - look at the first item of the list and return its value.
  - `isEmpty()` - returns boolean `true` if head node is null; `false` otherwise.
  - `length()`: returns as an `int` the number of commands in the stack.
  - `dump()`: prints the contents of the stack to standard output.
2. **The Node:** define a class called `Node` for the nodes in your `Stack`. It can either be an external class in a separate file called `Node.java` or an inner class of `Stack`. It should have the following:
  - A member variable to hold the string value.
  - A member variable to hold a link to another `Node`.
  - A constructor that takes a value as a string argument and copies that value into the `Node`'s private member variable.

You may have additional member variables and methods if they are useful to you, but they should be private.

#### Part Two:

Create a new Java class called `Parser` in a file called `Parser.java`. In this file you are going to write a public method called `check()` which will use your `Stack` class to determine if a string is well-formatted with regards to its parentheses. For example, given the input: `()()`, your program should return `"true"` as there are the correct number of brackets in the correct order. Alternatively, given the invalid input: `()() (`, your program should return `"false"` as there are the incorrect number of brackets. Likewise, given the invalid input: `) ) () ( (`, your program should return `"false"` as, although there are the correct number of brackets, they are not in the correct order.

In your `Parser` class write a method called `getString()` which will generate random strings with random parentheses. Your function should generate correctly and incorrectly formatted equations. All equations should be no longer than 10 characters. Use this method to debug your `check()` method.

A program will be posted on Moodle about a week before the assignment is due. Ensure that your classes work with this program as this will be used during marking.

### **Assessment:**

Completing Part One can earn up to a B+ grade, but to be eligible for an A+ you must also complete Part Two. Your solution will be marked on the basis of how well it satisfies the specification, how well-formatted and easy to read your code is, and whether each class and public method has at least some comment explaining what it does, what it's for, and what any of its arguments are (i.e. documentation).

Your code should compile and run as a console program from the Linux command-line (i.e. no GUI). Students are encouraged to test their code in the lab prior to submitting their solutions.

### **Submission:**

Create an empty directory (i.e. folder) using your student ID number as the directory name. Place copies of your source code (.java files) in this directory. If you wish to communicate with the marker any additional information then you may include a plain text README file, but nothing else (e.g. no compiled code). Upload this directory through the Moodle submission page for this assignment.