Institut für Informatik — TU München
Scientific **C**omputing in **C**omputer **S**cience
K. Rahnema, M. Schreiber

WS 2012

Tutorial
(Advanced Programming)
Worksheet 10:

Assignment 1: A more sophisticated Calculator

In this Worksheet we want to re-program the simple-calculator from Worksheet 3 to accept full mathematical expressions and calculate the final result. To do so, we need to implement two classes:

- A class which reads the input expression and breaks it down into simple mathematical expressions with only one operator.
- A class which receives the simple mathematical expressions, calculates the results and returns it.

There are many ways to implement the input reader class. One in particular could be breaking the the expressions into a binary tree where every inner node holds one mathematical expression and the leave nodes are the values. Consider the following example:

$((a + b) * c)$

Here we have got an expression with 2 operators and 3 values. The input reader class has to break this expression into 2 parts.

- $(x * c)$
- $x = (a + b)$

The first expression would create the first level of the binary tree using the $*$ operator as the operator held at the root node and x and c could now be either a value or another expression. If they are values they will be directly stored as a child node for the root, else, they will again be broken down into simpler expression until the tree is complete. A bottom-up traversal for the tree would now calculate the full expression.

The SimpleMath class will calculate each of these sub-expressions and return the corresponding value.

Questions:

- How does this tree structure assure that there will be no unbalanced parentheses in the input?