CS 130 Theory of Computation **2nd Semester, SY 2016-17**
Project 2

Recursive Descent Parser for Arithmetic Calculations

This project builds on the lexical analyzer of Project 1. Write a predictive parser for a simple calculator language. Your parser will be implemented through recursive procedures as discussed in class. The input will be a source program and the output will be the corresponding result of each command in the program. Consider the following sample input code, to demonstrate what you need to support:

```
// this program calculates the roots of a quadratic equation
disc = 10**2 - (4*5.5*(-3)); # compute discriminant
IF (disc >= 0) root1 = (-10 + SQRT(disc))/(2*5.5);
IF (disc >= 0) root2 = (-10 - SQRT(disc))/(2*5.5);
PRINT(root1);
PRINT(root2);
IF (disc < 0) PRINT("no real roots");
PRINT("thank you");
# end of the program
```

the following output should result:

```
computation performed (disc = 166.00)
condition met, computation performed (root1 = 0.26)
condition met, computation performed (root2 = -2.08)
output (0.26)
output (-2.08)
condition not met
output (thank you)
```

This simple calculator language has only two kinds of statements—assignment statements and print statements, both of which may be preceded by a conditional clause. Assignment statements take the form <id>=<expression>; while print statements take the form PRINT(<expression>); or PRINT(<string-literal>). Conditional clauses take a very simple form: IF (<expression> <rel-operator> <expression>).

Reserved words in this language are: SQRT, IF, and PRINT (case sensitive). You will need to revise your project 1 code to recognize the indicated reserved words as tokens distinguishable from identifiers, and add a few more (relational) operators (>, >=, <. <=, ==,!=) plus the semicolon token.

Assume all variables are of type double and allocate space for them upon first use. If a variable is used without a previous assignment, assume its value is 0. Finally, note that every statement results in some output, even if the statement includes a conditional.

Visit the course website for any updates and clarifications. This program is due on April 21 (midnight); submit your source code via moodle.