# CS3210, Summer 2013, Assignment 1

Create a recursive-descent parser in Java that parses the following language:

program -> statementList

statementList -> statement statementList | epsilon

statement -> "read" id | "write" id | forLoop | whileLoop | conditional | id ":=" expression

expression -> term operation term

term -> id | number

id -> "[a-zA-Z]+"

number -> “[0-9]+”

operation -> "+" | "-" | "/" | "\*"

forLoop -> "for" id number number statementList "rof"

whileLoop -> "while" condition statementList "elihw" |

conditional -> "if" condition statementList "fi" |

condition -> id comparison term

comparison -> "==" | "!=" | "<" | ">" | "<=" | ">="

Read from System.in. Your parser must be able to parse sufficiently complicated programs in this language such as

while a != b

if a > b

a := a − b

if a <= b

b := b − a

elihw

Please produce output that shows the parsing process. I recommend you use the Scanner class for lexical analysis. You probably should also become familiar with the String.matches() method.

Your program will be graded with 40% of the credit for design, 30% for whether it runs, and 30% for the testing you performed. For the design, include in comments in your single Java file, how you designed the code. At the top, provide the overall approach, and before each significant method, describe what the method does. Use Javadoc conventions for parameters, returned values, etc. Use inline comments for obscure code only. A decent guideline to commenting can be found at http://www.edparrish.net/common/javadoc.html though you are not required to follow it exactly. For the 30% test credit, again use comments at the bottom of the single Java file showing the tests you created and the results of your testing. Test for both good and bad input programs. Due June 10th, 2013, by midnight. A 10% per day late penalty will be accessed. You will be submitting you program online; details to follow.