# ICSI 311 Assignment 7 –Finish the Parser

**If you have bugs or missing features in this, you will need to fix them before you can continue on to new assignments. This is very typical in software development outside of school.**

**You must submit .java files. Any other file type will be ignored. Especially “.class” files.**

**You must not zip or otherwise compress your assignment. Blackboard will allow you to submit multiple files.**

***You must submit buildable .java files for credit.***

You will submit your existing files (Parser.java, Node.java, IntegerNode.java, FloatNode.java, MathOpNode.java, Basic.java, Token.java, Lexer.java, VariableNode.java, PrintNode.java, AssignmentNode.java, StatementNode.java, StatementsNode.java, StringNode.java, DataNode.java, ReadNode.java, InputNode.java, LabeledStatementNode.java, GosubNode.java, ReturnNode.java, ForNode.java, NextNode.java) and your new files: IfNode.java, BooleanOperationNode.java, FunctionNode.java

Add statement for IF. Note that you will need to create a Boolean expression parser. We are going to keep this super simple – a Boolean expression must follow the pattern: Expression operator Expression

Operator is >, >=, <, <=, <>, =

Finally, we need to deal with functions. Add the function names to the lexer. Functions are an additional option for EXPRESSION. Remember that we had EXPRESSION as

EXPRESSION = TERM { (plus or minus) TERM}

It now should be:

EXPRESSION = TERM { (plus or minus) TERM} or functionInvocation

Make a new parser method for functionInvocation that looks for the appropriate keywords and returns a FunctionNode or NULL. A function follows this pattern:

FUNCTIONNAME LPAREN parameterList RPAREN

A parameterList is empty OR a list (1 or more) of EXPRESSION or STRING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rubric | Poor | OK | Good | Great |
| Comments | None/Excessive (0) | “What” not “Why”, few (5) | Some “what” comments or missing some (7) | Anything not obvious has reasoning (10) |
| Variable/Function naming | Single letters everywhere (0) | Lots of abbreviations (5) | Full words most of the time (8) | Full words, descriptive (10) |
| Create the AST classes | None (0) | Classes missing (5) | All classes present, some methods missing (10) | All classes and methods (20) |
| Handle Boolean expressions | None (0) |  | Significantly Attempted (10) | Correct (20) |
| Handle If | None (0) |  | Significantly Attempted (10) | Correct (20) |
| Handle Functions | None (0) |  | Significantly Attempted (10) | Correct (20) |