Thomas Ferdinand

tferdina@students.kennesaw.edu

Kennesaw State University

Concepts of Programming Lang CS4308 Section 94

project deliverable 3

# Initial Problem Statement

3. **Module\_7 – 3rd Deliverable**

Develop a complete interpreter that executes with the scanner and parser. You must show the execution of this program by using a relevant source file as input, the program must show the results after executing the statements. Write a short report describing the work performed and include the input file and output data produced.

# Summary and Purpose of the Assignment

The third deliverable finished the program by executing the statements recognized by the parser. These statements include arithmetic and Boolean expressions. They can either be part of assignment statements or separate statements.

# Description of the Solution Used in the Project

The parser creates expressions as it catches them by reading tokens ahead of the current token and determining whether they create an expression. When an expression is detected, an object is created of the expression and the result is calculated. The expression object is then added to an ArrayList. The expressions are executed when they are detected. A row and column tracker was implemented to track where

# Table or Lists of Results

The data in the input file is meant to test the algorithms implemented by the program, such as assigning values and recognizing expressions. The output file also lists each expression and its result. The output file is appropriately named “output.txt”. A full table of results will be included at the end of the report.

The program is ready to run as is with the default input file ready to go. Changes can be made to the input file, but it must always be named “input.txt”

# Program Specifications

## Lexical Rules :

identifier → letter char\_list

char\_list → letter\_digit char\_list | letter\_digit

literal\_integer → digit literal\_integer | digit

assignment\_operator → =

le\_operator → <=

lt\_operator → <

ge\_operator → >=

gt\_operator → >

eq\_operator → = =

ne\_operator → ~=

add\_operator → +

sub\_operator → -

mul\_operator → \*

div\_operator → /

## IDE used:

IntelliJ IDEA 2016.2.1

Build #IC-162.1447.26, built on August 2, 2016

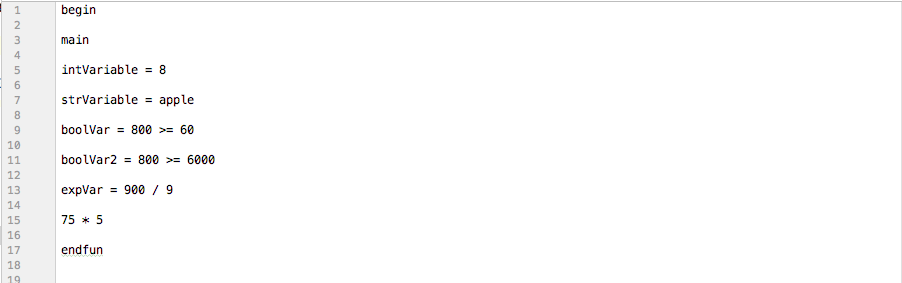
JRE: 1.8.0\_76-release-b216 x86\_64

JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

# Limitations of Program

The program does not execute expressions that include string variables holding values. The program can, however, assign the result of an expression to a variable. Only the above characters are recognized by the program. All other characters are treated as “invalid characters” and will cause errors/exceptions.

# Input and Results

Input File:

# Input and Results

Output File:

