CS 345 - Programming Languages
Assignment 3

1 Extending mySIMPL (30 Points)

The task is to extend the parser and evaluator for mySIMPL (my Simple IMperative Programming Language) from assignment 2. In this assignment we add support for functions, conditional statement, and loops.

The extended syntax (bold font used to highlight changes to the previous grammar) of mySIMPL is:

```
prog ::= retStatement '.' | funcDecl ';' prog | statement ';' prog
funcDecl ::= 'function' < id > '()' '<' prog '>' '{' prog '}'
retStatement ::= 'return' base
statement ::= declaration | assignment | declAssignment | conditional | loop
declaration ::= < id > '=' base
assignment ::= < id > '=' base
declAssignment ::= < id > '=' base
conditional ::= 'if' '(' condition ')' 'then' statementSeq [ 'else' statementSeq ] 'endif'
loop ::= 'while' '(' condition ')' 'do' statementSeq 'done'
statementSeq ::= statement '.‐' | statement ';' statementSeq
condition ::= base comp base | '(' condition logOp condition ')' | boolean | '!' '(' condition ')
boolean ::= 'true' | 'false'
base ::= < id > | < number > | '(' expression ')' | funcCall
funcCall ::= < id > '(' base ')'
expression ::= [expression addOp] term
term ::= [term mulOp] factor
factor ::= base
addOp ::= '+' | '‐'
mulOp ::= '*' | '/'
logOp ::= '&&' | '||'
comp ::= '==' | '<' | '>' | '<=' | '>=' | '!='
```
Extend the predicates

\[\text{parse}(+\text{TokenList}, -\text{AST})\]
\[\text{evaluate}(+\text{AST}, -\text{Number})\]

according to the new grammar, static scoping rules, and call-by-value semantics.

Example programs are:

\[
[\text{'var'}, x, '\leftarrow', 1, '；', '\text{if}', '\{', x, '\leftarrow', 0, '}', '\text{then}', x, '\leftarrow', 10, '；', '\text{else}', x, '\leftarrow', 20, '；', '\text{endif}', '；', '\text{return}', x, '．']
\]

\[
[\text{function}, f, '\{', x, '\}', '\{', '\text{return}', x, '．', '}', '；', '\text{return}', f, '\{', '\{', '10, \text{+}, 1, '}', '\}', '．']
\]

2 Advanced Loop Control Flow (Optional, 10 Bonus Points)

Many commonly-used programming language support changing the control flow in loops through instructions like \text{break} and \text{continue}. Implement support for the two instructions for mySIMPL. (Additional information will soon be given on Piazza.)

Additional Information and Hints

- All single-quoted terminals in the grammar should be considered keywords and hence excluded as identifiers. Numbers and the boolean values \text{true} and \text{false} are excluded from being identifiers.
- Albeit expressed through the same element in the grammar, identifiers for functions and variables should be considered separate namespaces and therefore a function and a variable of the same name should be allowed to coexist within a program.
- One of the key challenges of this assignment is to deal with the scoping and visibility of variables in blocks. E.g., declarations within function bodies should not leak into the enclosing scope.
- We will expect function declarations to always precede any corresponding function call even though the grammar does not enforce this. Therefore, it is safe to use a single pass for evaluation.