SEAN CONNOR
24 JULY 2018

Compiler Top-Down Parse Exercise

Summer - 2018

Given this Top-Down parse grammar:

1. <stmt-list> ::= <stmt> { ; <stmt> }
2. <stmt> ::= <assign> | <read> | <write> | <for>
3. <assign> ::= **id** := <exp>
4. <exp> ::= <term> { + <term> | - <term> }
5. <term> ::= <factor> { * <factor> | DIV <factor> }
6. <factor> ::= **id** | **int** | (<exp>)
7. <read> ::= READ (<id-list>)
8. <id-list> ::= **id** { , **id** }
9. <write> ::= WRITE (<id-list>)
10. <for> ::= FOR <index-exp> DO <body>
11. <index-exp> ::= **id** := <exp> TO <exp>
12. <body> ::= <stmt> | BEGIN <stmt-list> END

Draw the Top-Down parse tree for this <stmt>

```
FOR N := 1 TO 20 DO
        BEGIN
            READ ( value );
            sumSQ := sumSQ + value * value;
            sum   := sum + value
        END
```

- See attached for tree

SEAN CONNOR
24 July 2018

Bottom-Up Parse Workshop - Spring 2018

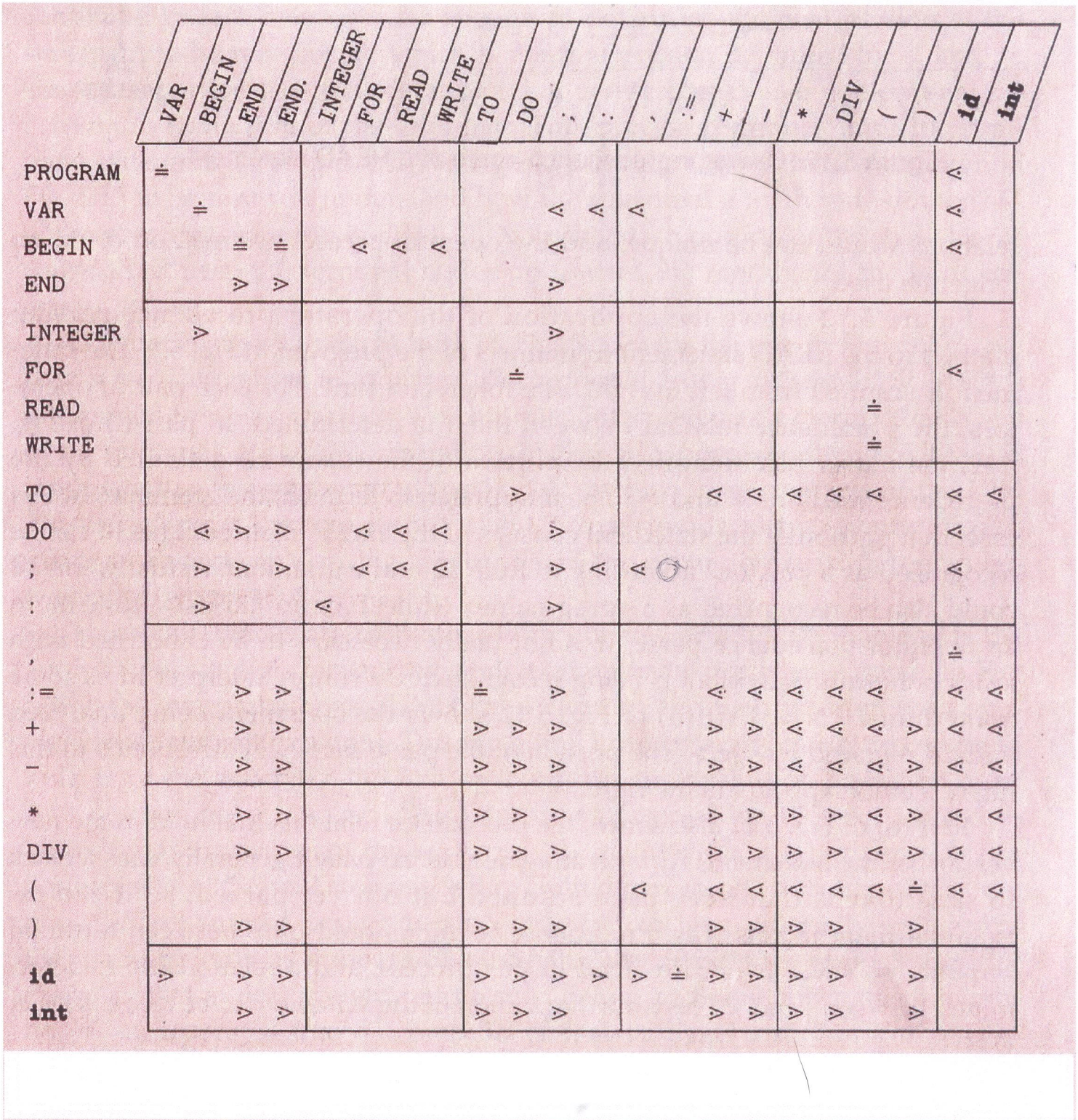
Draw the Parse tree for this **<statement-list>** using the Compiler Precedence Matrix, and the Top - Down Parse Grammar.
The tree should grow up toward the top of the page.
Insert the left / center / right parse precedence markers as you draw the tree. <· ·= ·>
The parsing is left to right.

- See attached for tree
- See below for precedence markers

mean := sum DIV 20 ; variance := sumSQ DIV 20 - mean * mean ; WRITE (mean , variance)

<· <· ·>·<· <· ·> <· ·>·<· ·=<· ·= ·> ·>

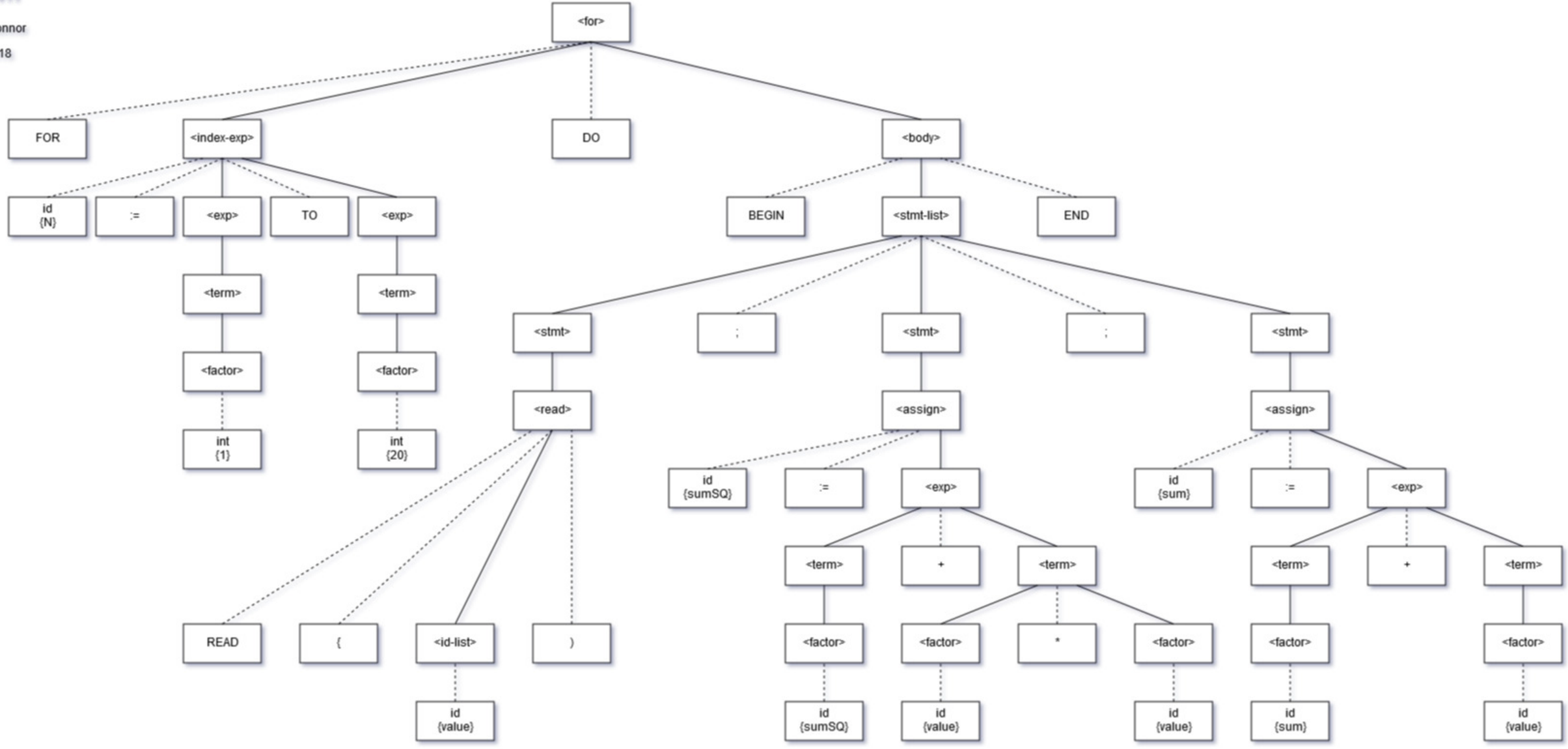
SEAN CONNOR
24 JULY 2018



Top-Down

@author Sean Connor

@date 24 July 2018



Bottom-Up

@author Sean Connor

@date 24 July 2018