

```
1  /****
2  *
3  * This is the CUP specification for a simple expression list grammar, as
4  * discussed in CSC 330 Lecture Notes Week 3. This grammar is quite similar to
5  * the expression list example in Section 1 of the CUP User's Manual.
6  *
7  */
8
9  import java_cup.runtime.*;
10
11 /* Terminals (tokens returned by the lexer). */
12 terminal      SEMI, PLUS, MINUS, TIMES, DIVIDE;
13 terminal String VAR;
14
15 /* Non terminals */
16 nonterminal TreeNode      expr;
17 nonterminal TreeNodeList  expr_list;
18
19 /* Precedences */
20 precedence left PLUS, MINUS;
21 precedence left TIMES, DIVIDE;
22
23 /* The grammar */
24 expr_list ::= expr:e
25           { : RESULT = new TreeNodeList(e, null); :}
26           | expr:e SEMI expr_list:e1
27           { : RESULT = new TreeNodeList(e, e1); :}
28           ;
29
30 expr      ::= expr:e1 PLUS expr:e2
31           { : RESULT = new TreeNode2(sym.PLUS, e1, e2); :}
32           | expr:e1 MINUS expr:e2
33           { : RESULT = new TreeNode2(sym.MINUS, e1, e2); :}
34           | expr:e1 TIMES expr:e2
35           { : RESULT = new TreeNode2(sym.TIMES, e1, e2); :}
36           | expr:e1 DIVIDE expr:e2
37           { : RESULT = new TreeNode2(sym.DIVIDE, e1, e2); :}
38           | VAR:v
39           { : RESULT = new LeafNode(sym.VAR, v); :}
40           ;
```