

## UE20CS353-CD LAB-4

NAME	SRN	CLASS & SECTION
VIJAY J	PES2UG20CS815	6 - J

### Lexer.l

```
1 %{
2 // NAME: Vijay J
3 // SRN: PES2UG20CS815
4 // SECTION: J
5 #define YYSTYPE char*
6 #include <unistd.h>
7 #include "y.tab.h"
8 #include <stdio.h>
9 extern void yyerror(const char *); // declare the error handling function
10 %}
11
12 /* Regular definitions */
13 digit [0-9]
14 letter [a-zA-Z]
15 id {letter}({letter}|{digit})*
16 digits {digit}+
17 opFraction (\.{digits})?
18 opExponent ([Ee][+-]?{digits})?
19 number {digits}{opFraction}{opExponent}
20 %option yylineno
21
22 %%
23 \\/(.*) ; // ignore comments
24 [\t\n] ; // ignore whitespaces
25 "(" {return *yytext;}
26 ")" {return *yytext;}
27 "." {return *yytext;}
28 "," {return *yytext;}
29 "*" {return *yytext;}
30 "+" {return *yytext;}
31 ";" {return *yytext;}
32 "-" {return *yytext;}
33 "/" {return *yytext;}
34 "=" {return *yytext;}
35 ">" {return *yytext;}
36 "<" {return *yytext;}
37 {number} {
38     yylval = strdup(yytext); //stores the value of the number to be used
    later for symbol table insertion
39     return T_NUM;
40 }
41 {id} {
42     yylval = strdup(yytext); //stores the identifier to be
    used later for symbol table insertion
43     return T_ID;
44 }
45 . {} // anything else => ignore
46 %%
```

# Parser.y

```
1 %{\n2 // NAME : Vijay J\n3 // SRN: PES2UG20CS815\n4 // SECTION: J\n5     #include "abstract_syntax_tree.c"\n6     #include <stdio.h>\n7     #include <stdlib.h>\n8     #include <string.h>\n9     void yyerror(char*\n\ns);\nerror handling function\n10     int\n11     yylex();\ndeclare the function performing lexical analysis\n12     extern int\n13     yylineno;\n14     track the line number\n15 %}\n16\n17 %union\n18 union to allow nodes to store return different datatypes\n19 {\n20     char* text;\n21     expression_node* exp_node;\n22 }\n23\n24 %token <text> T_ID T_NUM\n25\n26 %type <exp_node> E T F
```

```
27\n28 /* specify start symbol */\n29 %start START\n30\n31 %%\n32 START : ASSGN    {\n33     printf("Valid syntax\\n");\n34     YYACCEPT;\n35     program fits the grammar, syntax is valid\n36 }\n37\n38 /* Grammar for assignment */\n39 ASSGN : T_ID '=' E    {\n40     // displaying the expression tree\n41     printf("\\nExpression Tree:\\n");\n42     display_exp_tree($3);\n43 }\n44\n45 ;\n46\n47 /* Expression Grammar */\n48 E : E '+' T    {\n49     // create a new node of the AST and set left and\n50     right children appropriately\n51     $$ = init_exp_node(strdup("+"), $1, $3);\n52 }\n53\n54 | E '-' T    {\n55     // create a new node of the AST and set left and\n56     right children appropriately\n57     $$ = init_exp_node(strdup("-"), $1, $3);\n58 }\n59\n60 | T    { $$ = $1; }
```

```
Open  ▾  *parser.y  Save  -  □  ×
~/PESU/6th Sem/CD/CD LAB/Week - 4/Lab4

54      | ' '  {
55      ;
56
57
58 T : T '*' F      {
59     right children appropriately          // create a new node of the AST and set left and
60                                         $$ = init_exp_node(strdup("*"), $1, $3);
61                                         }
62     | T '/' F      {
63     right children appropriately          // create a new node of the AST and set left and
64                                         $$ = init_exp_node(strdup("/"), $1, $3);
65                                         }
66     | F {
67         //pass AST node to the parent
68         $$ = $1;
69     }
70 ;
71
72 F : '(' E ')' { $$ = $2; }
73     | T_ID {
74         // creating a terminal node of the AST
75         $$ = init_exp_node(strdup($1), NULL, NULL);
76     }
77     | T_NUM {
78         // creating a terminal node of the AST
79         $$ = init_exp_node(strdup($1), NULL, NULL);
80     }
81 ;
82
83 %%
84
```

```
85
86 /* error handling function */
87 void yyerror(char* s)
88 {
89     printf("Error :%s at %d \n",s,yylineno);
90 }
91
92 int yywrap() {
93     return 1;
94 }
95
96 /* main function - calls the yyparse() function which will in turn drive yylex() as well */
97 int main(int argc, char* argv[])
98 {
99     yyparse();
100    return 0;
101 }
```

## Abstract\_syntax\_tree.c

```
abstract_syntax_tree.c
~/PESU/6th Sem/CD/CD LAB/Week - 4/Lab4

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include "abstract_syntax_tree.h"
5
6 expression_node* init_exp_node(char* val, expression_node* left, expression_node* right)
7 {
8     // function to allocate memory for an AST node and set the left and right children of
the nodes
9     expression_node* new_ = (expression_node*)malloc(sizeof(expression_node));
10    new_>left = left;
11    new_>right = right;
12    new_>value = (char*)malloc(sizeof(val) + 1);
13    strcpy(new_>value, val);
14    return new_;
15 }
16
17 void display_exp_tree(expression_node* exp_node)
18 {
19     // traversing the AST in preorder and displaying the nodes
20     if(exp_node != NULL)
21     {
22         printf("%s\n", exp_node->value);
23         display_exp_tree(exp_node->left);
24         display_exp_tree(exp_node->right);
25     }
26 }
```

## Makefile

```
makefile
~/PESU/6th Sem/CD/CD LAB/Week - 4/Lab4

1 compiler: y.tab.c lex.yy.c
2 gcc y.tab.c lex.yy.c
3
4 y.tab.c: parser.y
5 yacc -d parser.y
6
7 lex.yy.c: lexer.l
8 lex lexer.l
9
10 clean:
11 rm *.yy.c y.tab.h y.tab.c a.out
```

## Input And Output

### Test\_input\_1.c

```
test_input_1.c
~/PESU/6th Sem/CD/CD LAB/Week - 4/Lab4

1 a = 10 / 5 - 2 * 7 + 3
```

## Output\_1

```
~/P/6/C/C/P/Lab4

yoyo@zaemon in ~/PESU/6th Sem/CD/CD LAB/PES2UG20CS815/Lab4 via C v12.2.1-gcc took 2ms
λ make
yacc -d parser.y
lex lexer.l
gcc y.tab.c lex.yy.c

yoyo@zaemon in ~/PESU/6th Sem/CD/CD LAB/PES2UG20CS815/Lab4 via C v12.2.1-gcc took 317ms
λ ./a.out < test_input_1.c

Expression Tree:
+
-
/
10
5
*
2
7
3
Valid syntax

yoyo@zaemon in ~/PESU/6th Sem/CD/CD LAB/PES2UG20CS815/Lab4 via C v12.2.1-gcc took 1ms
λ _
```

## Test\_input\_2.c

```
test_input_2.c
~/PESU/6th Sem/CD/CD LAB/Week - 4/Lab4

1 b = c / 6.7 + 12.45 - a * 1234.0
```

## Output\_2

```
~/P/6/C/C/P/Lab4

yoyo@zaemon in ~/PESU/6th Sem/CD/CD LAB/PES2UG20CS815/Lab4 via C v12.2.1-gcc took 2ms
λ make
yacc -d parser.y
lex lexer.l
gcc y.tab.c lex.yy.c

yoyo@zaemon in ~/PESU/6th Sem/CD/CD LAB/PES2UG20CS815/Lab4 via C v12.2.1-gcc took 323ms
λ ./a.out < test_input_2.c

Expression Tree:
-
+
/
c
6.7
12.45
*
a
1234.0
Valid syntax

yoyo@zaemon in ~/PESU/6th Sem/CD/CD LAB/PES2UG20CS815/Lab4 via C v12.2.1-gcc took 1ms
λ _
```