

## WEEK-4: Design LALR bottom up parser for a given language.

### Week4.l

```
%{
#include<stdio.h>
#include "y.tab.h"
}%
%%
[0-9]+ {yylval.dval=atof(yytext);
return DIGIT;
}
\n|. return yytext[0];
%%
```

### Week4.y

```
%{
/*This YACC specification file generates the LALR parser for the program
considered in experiment 4.*/

#include<stdio.h>

}%

%union
{
double dval;
}

%token <dval> DIGIT

%type <dval> expr
%type <dval> term
%type <dval> factor

%%

line: expr '\n' {
printf("%g\n", $1);
}
;

expr: expr '+' term {$$=$1 + $3 ;}
| term
;

term: term '*' factor {$$=$1 * $3 ;}
```

```

| factor
;
factor: '(' expr ')' {$$=$2 ;}
| DIGIT
;
%%

int main()
{
yyparse();
}

yyerror(char *s)
{
printf("%s",s);
}

```

```

vnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC: ~/Desktop/kk
vnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC:~/Desktop/kk$ lex week4.l
vnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC:~/Desktop/kk$ yacc -d week4.y
vnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC:~/Desktop/kk$ cc lex.yy.c y.tab.c -ll -lm
week4.y: In function 'main':
week4.y:31:1: warning: implicit declaration of function 'yyparse' [-Wimplicit-function-declaration]
   31 | yyparse();
      | ~~~~~~
week4.y: At top level:
week4.y:33:1: warning: return type defaults to 'int' [-Wimplicit-int]
   33 | yyerror(char *s)
      | ~~~~~~
y.tab.c: In function 'yyparse':
y.tab.c:182:23: warning: implicit declaration of function 'yylex' [-Wimplicit-function-declaration]
   182 |         if ((yychar = yylex()) < 0) yychar = 0;
       |                       ~~~~~~
vnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC:~/Desktop/kk$ ./a.out
2+3
5
syntax errorvnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC:~/Desktop/kk$ ./a.out
4*5
20
syntax errorvnrvjlet@vnrvjlet-HP-ProDesk-400-G7-Microtower-PC:~/Desktop/kk$ 

```