# Department of Computer Science and Engineering

# Compiler Design Lab (CS 306)

## Week 9: Implementation of LALR parser using LEX and YACC

## Week 9 Programs

1. Implement LALR parser using LEX and YACC for the following Grammar:

   E → E+T |T  
   E’→ T*F | F  
   F → (E) | d

2. Implement LALR parser using LEX and YACC for the following Grammar by specifying proper precedence for operators:

   E → E+E | E-E | E*E | E/E | -E | (E) | digit

## Instructions:

- Explanation and code of first program is given below.
- YouTube link of this week’s explanation is https://www.youtube.com/watch?v=yKFfqthNsE0
- Implement both programs and upload into your Github accounts under the folder **Week9-Lab-exercise**

## Program:

Step 1: Open a text file using notepad, name it parser.l and write the lex code in it.

Code in parser.l

```
%{
#include "y.tab.h"
%}

%%
[0-9]+  {yylval=atoi(yytext);
          return NUMBER;
         }
[\t]      ;
\n       return 0;
.        return yytext[0];
%%
```

**Step 2:** Open a text file using notepad, name it parser.y and write the yacc code in it.

Code in parser.

```
%{
 #include<stdio.h>
%}

%token  NUMBER

%%
S: E                    { printf("The result is =%d\n",$1);}
  ;
E: E'+'T                { $$ = $1 + $3; }
   | T                  { $$ = $1;}
   ;
T: T'*'F                { $$ = $1 * $3; }
   | F                  { $$ = $1;}
   ;
F: '('E')'              { $$ = $2;}
  | NUMBER              { $$ = $1;}
  ;
%%

int main(){
  yyparse();
}

int yywrap(){
    return 1;
}

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

**Step 3:** Open command prompt (Windows button -> run ->cmd )

**Step 4:** Go to the folder in which your files are saved

**Step 5:** Type the following commands

```
C:/>flex parser.l
C:/>yacc -d parser.y
C:/> gcc lex.yy.c y.tab.c -w
C:/>a
```

Type string and get your output.