Download Flex from - https://sourceforge.net/projects/gnuwin32/files/flex/2.5.4a-1/flex-2.5.4a-1.exe/download?use_mirror=cyfuture&download=

Download Bison From –

https://sourceforge.net/projects/gnuwin32/files/bison/2.4.1/bison-2.4.1-setup.exe/download?use_mirror=cyfuture

Download Mingw C compiler by Following the tutorial : https://www.youtube.com/watch?v=JsO58opI3SQ

Step – 1 These Both will be downloaded at default storage in common folder at :

C:\Program Files (x86)\GnuWin32

Step 2: Move the GnuWin32 folder from Program Files (x86) to C drive folder Now the path must be :   C:\GnuWin32

Step 3 : Open Environment Variable and add a path "C:\GnuWin32\bin"

Step 4 : Restart the computer

Step 5 : Check following three commands in command prompt one by one to check successful installation

Flex --version

Bison --version

Gcc --version

Step 6: Now Create a new folder in you desired path and open that in vs code

Step 7 : Create a new file in that folder named calc.l and add the following code

```
%{
#include "calc.tab.h"  // Include Bison header file
%}

%%
[0-9]+(\.[0-9]+)? {
    yylval.num = atof(yytext);  // Convert matched text to a double
    return NUMBER;
}

[ \t]    ;   // Ignore whitespace
\n       return '\n';  // Newline character
.       return yytext[0];  // Return any other character
%%

int yywrap(void) {
    return 1;
}
```

Step 8: Create a file named calc.y and paste the following code

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// Function prototype for error handling
void yyerror(const char *s);
int yylex(void);
%}

%union {
    double num;  // For numerical values
}

// Declare tokens and their associated types
%token <num> NUMBER

// Associate non-terminal symbols with a data type
%type <num> expr
%left '+' '-'
%left '*' '/'
%%

// Grammar rules and actions
input:
    /* empty */
    | input line
    ;
line:
    expr '\n' { printf("Result: %lf\n", $1); }
    ;
expr:
    expr '+' expr { $$ = $1 + $3; }
    | expr '-' expr { $$ = $1 - $3; }
    | expr '*' expr { $$ = $1 * $3; }
    | expr '/' expr {
        if ($3 == 0) {
            yyerror("Division by zero");
            $$ = 0;
        } else {
            $$ = $1 / $3;
        }
    }
    | '(' expr ')' { $$ = $2; }
    | NUMBER { $$ = $1; }
    ;
%%
void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}

int main() {
    printf("Enter an expression (e.g., 3 + 4 * 2):\n");
    return yyparse();
}
```

Step 9 : Go to terminal of VS Code and Ensure the path is referring your Folder

Step 10 : Run the following command in proper order one by one

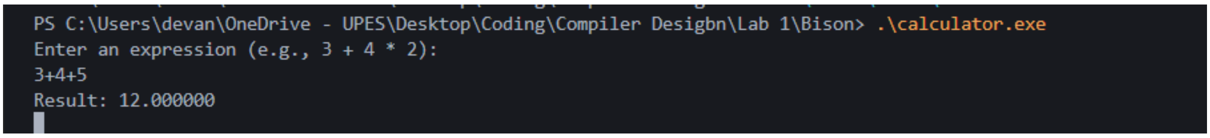
bison -d calc.y

flex calc.l

gcc calc.tab.c lex.yy.c -o calculator

.\calculator.exe

Output must be like this



```
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 1\Bison> .\calculator.exe
Enter an expression (e.g., 3 + 4 * 2):
3+4+5
Result: 12.000000
```