LAB REPORT

CSE232: Compiler Design Lab

03
[Report Number]

Topic: Write a calculator using Flex and Bison.

Submitted To

Mushfiqur Rahman Chowdhury (MRC)

Lecturer

Department of CSE, Daffodil International University

Submitted By

Student ID: 024222005101781

Section: 63 N

Student Name: SHAHARIAR HOSSAIN

Date of Assignment Distribution: 30 November 2025

Date of Assignment Submission: 11 November 2025

Experiment No: 03	Mapping: CO1 and CO2

Experiment Name	Write a calculator using Flex and Bison.

Experiment Details:

Write a calculator using Flex and Bison that supports:

1. Basic arithmetic operations (+,-,/,*)

2. Parentheses

3. Make sure that the operations happen in the proper order.

Lexer.l:

```
%{
#include "parser.tab.h"
#include <stdio.h>
#include <stdlib.h>
%}

%%

[0-9]+(\.[0-9]+)?    { yylval.d = atof(yytext); return NUMBER; }
[ \t]+               { /* skip whitespace */ }
\n                   { return '\n'; }
"+"                  { return '+'; }
"-"                  { return '-'; }
"*"                  { return '*'; }
"/"                  { return '/'; }
"("                  { return '('; }
")"                  { return ')'; }
.                    { fprintf(stderr, "Unknown character: %s\n", yytext); }

%%

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

Parser.y:

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

void yyerror(const char *s);
int yylex(void);
%}
```

```
%union {
    double d;
}

%token <d> NUMBER
%left '+' '-'
%left '*' '/'
%right UMINUS

%type <d> expr

%%

input:
    /* empty */
  | input line
;

line:
    '\n'
  | expr '\n' { printf("Result: %g\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; }
  | '-' expr %prec UMINUS { $$ = -$2; }
  | NUMBER { $$ = $1; }
;

%%

extern FILE *yyin;

void yyerror(const char *s) { fprintf(stderr, "Error: %s\n", s); }

int main(int argc, char *argv[]) {
    if (argc > 1) {
        FILE *file = fopen(argv[1], "r");
        if (!file) {
            fprintf(stderr, "Error: Cannot open file %s\n", argv[1]);
```

```
            return 1;
        }
        yyin = file;
    }
    return yyparse();
}
```

tests.txt:

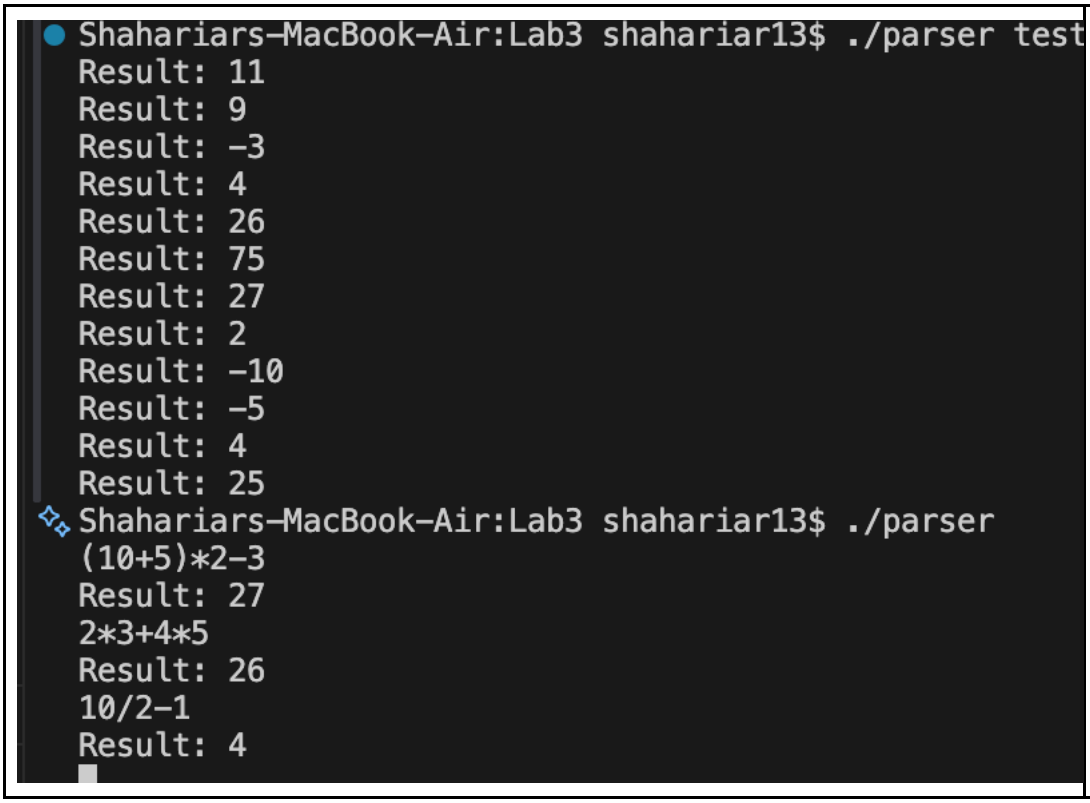
```
3+4*2
(1+2)*3
-5+2
10/2-1
2*3+4*5
100-50+25
(10+5)*2-3
12/3/2
5*-2
-(3+2)
1.5+2.5
(100/10)*2+5
```

Command:

```
flex lexer.l
clang lex.yy.c -o lexer
./parser tests.txt
./parser
```

Obtained Output:

	Desired Output?



YES

Alternative Steps/Solution (If any):

None.

Observation/ Comments:

I have implemented a Flex lexer and a Bison parser to create a simple calculator that supports basic arithmetic operations (+, -, *, /), parentheses, and respects operator precedence.

Appendix A: Course Outcomes, Complex Engineering Problems (EP) and Complex Engineering Activities (EA) Addressing.

Table: CSE312 Course Outcomes (COs) with Mappings

COs	CO Statements	POs	Learning Domains	Knowledge Profile	Complex Engineering Problem	Complex Engineering Activities

CO 1	Demonstrate a comprehensive understanding of fundamental database management concepts, including the relational data model, normalization techniques, and SQL basics.	PO1	C2 A2 P2	K2 K3 K4 K8	EP1 EP4	
CO 2	Design, implement and optimize relational databases, incorporating advanced SQL queries, indexing techniques and query optimization strategies.	PO3	C3 A3 P3	K2 K3 K4 K6 K8	EP1 EP2 EP7	EA3
CO 3	Understand and Analyze security measures, distributed database architectures and emerging trends in database management, demonstrating an understanding of the broader context and challenges in the field.	PO5	C4 A4 P3	K6	EP4	

Table: Addressing CO (1 to 3), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP):

SN	Engineering Problem (EP) Definition	Attain Ment	CO	Justification (with Knowledge Profile)
01	EP1: Depth of Knowledge required	Yes/No	CO1, CO2	
02	EP2: Range of Conflicting Requirements	Yes/No	CO2	
03	EP4: Familiarity of Issues	Yes/No	CO1, CO3	

04	EP7: Interdependence	Yes/No	CO2	

Table: Addressing COs

SN	COs	Attainment	Justification
01	CO1	Yes/No	These Lab activities attain CO1 by.
02	CO2	Yes/No	N/A
03	CO3	Yes/No	These Lab activities attain CO3 by.

Table: Lab-Wise Recommended Topics

Lab Class No.	**Proposed Activity**
Lab 1	Write a Flex program to recognize the following types of strings.
Lab 2	Write a Flex program to recognize the following types of strings.
Lab 3	Write a calculator using Flex and Bison.