**CD LAB VIVA QUESTIONS**

EXPERIMENT 1

1. Explain LEX and YACC tools briefly.
2. Give the structure of the LEX program.
3. Explain the structure of a YACC program.
4. What are tokens or terminal symbols?
5. What is lexical analyzer?
6. Discuss about the input buffering scheme in lexical analyzer.

EXPERIMENT 2

1. What are the limitations of CFG?
2. Explain Ambiguous and Unambiguous Grammar.
3. Which grammar can be translated to DFAs?
4. Differentiate between Ambiguous and Unambiguous Grammar.
5. Determine the language which is generated by the grammar

 S→aSa∣bSb∣a∣b over the alphabet of {a,b}

EXPERIMENT 3

1. How many Keywords are in C?
2. What is a C token and types of C tokens?
3. What is an Identifier?
4. How many types of operators are there in C?
5. What is a variable?

EXPERIMENT 4

1. Why is left recursion considered problematic?
2. Give an example of grammar with left recursion and show how to eliminate it.
3. How does left recursion affect parsing efficiency?
4. What is the difference between indirect and direct left recursion?
5. What is a recursion method?
6. Eliminate the left recursion for the following grammer

E-->E+T/T

T-->T\*F/F

F-->(E)/id

EXPERIMENT 5

1. Do left factoring in the following grammar-

S → a / ab / abc / abcd

1. What is the relationship between left recursion and left factoring?
2. [Why is left factoring used?](https://www.gatevidyalay.com/left-factoring-examples-compiler-design/)
3. How does left factoring help in parsing?
4. Provide an example of left factoring.

EXPERIMENT 6

1. When is the type checking usually done? (duringSDT)
2. Who checks every character of the source text in a Compiler? (LA)

For questions 3 and 4 refer to the data given below:

The programming language given below is written in the programming language that does not allow nested declarations of functions and allows global variables.  
global int j = 100, k = 5;  
void M(n)  
{  
int j = 10;  
print (n + 10);  
j = 200;  
k = 20;  
print (n);  
}  
main()  
{  
M(j + k);  
}

1. What is the output of the above program if the programming language uses static scoping and call by need parameter passing mechanism?
2. What is the output of the above program if the programming language uses dynamic scoping and call by name parameter passing mechanism?

EXPERIMENT 7

1. What is Bottom up parsing?
2. What do you mean by Top Down Parsing?
3. Differentiate between top down and bottom up parsing?
4. What is the leading of a non-terminal?
5. What is the trailing of a non-terminal
6. What are the techniques used in Bottom up Parsing?
7. Which parsing is more beneficial?

EXPERIMENT 8

1. What are four actions of Shift Reduce Parser?
2. Shift-reduce parsing is a form of bottom-up parsing. What is the purpose of the two actions, shift and reduce?
3. Describe the stack implementation of shift reduce parsing?
4. The Earley algorithm is an efficient context-free parsing algorithm which uses a chart data structure. Is this algorithm a top-down or a bottom-up algorithm?
5. State Error in each phase of compiling?

EXPERIMENT 9

1. Distinguish between FIRST and FOLLOW?
2. What is the significance of first & follow?
3. What do you mean by handle pruning?
4. What are the techniques used in Top Down Parsing?
5. How many rules are there to calculate First & Follow?
6. Given following grammar  
   S -> L=L  
   s -> L  
   L -> \*L  
   L -> id

What is the first and follow for the non-terminals?

1. If the grammar is changed into S -> L=R  
    S -> R  
    L -> \*R  
    L -> id  
    R -> L

What will be the first and follow?

EXPERIMENT 10

1. What do you mean by CFG?
2. What do you mean by operator precedence?
3. What is the property of Operator Precedence Grammar?
4. Why we require Operator Precedence Grammar?
5. What are Precedence Relations in Operator Grammar?
6. What is C Operator Precedence and Associativity?
7. What are Precedence Functions in compiler design?