ANURAG ENGINEERING COLLEGE

(An Autonomous Institution)

(IT504PC) COMPILER DESIGN LAB

III Year B.Tech. IT- I Sem

L	T	P	C
0	0	2	1

Prerequisites:

1. A Course on "Object Oriented Programming through Java".

Co-requisites:

1. A course on" Web Technologies".

Course Objectives:

The objectives of this course are to provide:

- To understand the various phases in the design of a compiler.
- To understand the design of top-down and bottom-up parsers.
- To understand syntax directed translation schemes.
- To understand storage allocation strategies
- To introduce yacc tools.

LIST OF EXPERIMENTS:

1. Implementation of symbol table.
2. Develop a lexical analyzer to recognize a few patterns inc (ex. Identifiers, constants, comments, operators etc.)
3. Implementation of lexical analyzer using lex tool.
4. Generate yacc specification for a few syntactic categories.
   a) Program to recognize a valid arithmetic expression that uses operator +,-, * and /.
   b) Program to recognize a valid variable which starts with a letter followed by any number of letter or digits.
   c) Implementation of calculator using lex and yacc.
5. Convert the bnf rules into yacc form and write code to generate abstract syntax tree.
6. Implement type checking
7. Implement any one storage allocation strategies (heap, stack, static)
8. Write a lex program to count the number of words and number of lines in a given file or program.
9. Write a 'C' program to implement lexical analyzer using c program.
10. write recursive descent parser for the grammar E->E+T E->T T->T*F T->F F->(E)/id.
11. write recursive descent parser for the grammar S->(L) S->aL->L,S L->S
12. Write a C program to calculate first function for the grammar
    E->E+T E->T T->T*F T->F F->(E)/id
13. Write a YACC program to implement a top down parser for the given grammar.
14. Write a YACC program to evaluate algebraic expression.

TEXT BOOK:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam,Ravi Sethi, Jeffry D. Ullman.

REFERENCE BOOKS:

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Louden, Thomson.

Course Outcomes:

1. Design, develop, and implement a compiler for any language.
2. Use lex and yacc tools for developing a scanner and a parser.
3. Design and implement LL and LR parsers.

CO-PO-PSO Mapping:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2
CO-1	H	H	M	H	H	M	L							H
CO-2	H	H	H	H	H	M	H							H
CO-3	H	H	M	H	H	M	M							H
CO-4	H	H	M	H	H	M	M							H
CO-5	L	M	M	H	H	M	M							H

H-HIGH **M**-MODERATE **L**-LOW