Compiler Design Lab

Course Code	19CS3651	**Year**	III	**Semester**	II
Course Category	Program Core	**Branch**	CSE	**Course Type**	Practical
Credits	1.5	**L-T-P**	0-0-2	**Prerequisites**	
Continuous Internal Evaluation :	25	**Semester End Evaluation:**	50	**Total Marks:**	75

Course Outcomes		
Upon successful completion of the course, the student will be able to		
CO1	Apply C, LEX and YACC programming to write a solution for the phases of compiler problems.	**L3**
CO2	Implement programs as an individual on different IDEs.	**-**
CO3	Develop an effective report based on various programs implemented.	**-**
CO4	Apply technical knowledge for a given problem and express with an effective oral communication.	**L3**
CO5	Analyze outputs generated by executing C, LEX and YACC programs for different test cases.	**L4**

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:Substantial, 2: Moderate, 1:Slight)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO1 0**	**PO1 1**	**PO1 2**	**PSO 1**	**PSO 2**
CO1	3											2		
CO2					3				3			3		
CO3										3				
CO4	3									3				
CO5		3												

Syllabus		
Expt. No.	**Contents**	**Mapped CO**
1	Design aLexicalanalyzerforthegivenlanguage. The lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value.	**CO1,CO2,CO3,CO4,CO5**
2	(a) Implement the lexical analyzer using LEX program for the regular expression RE's: a(a+b)* (b) Implement the LEX program to implement RE's: (a+b)*abb(a+b)*	**CO1,CO2,CO3,CO4,CO5**
3	(a) Implement the lexical analyzer using JLEX, FLEX or LEX or other lexical analyzer generating stools. (b) Implement the lexical analyzer Program to count no of +ve and –ve integers using LEX	**CO1,CO2,CO3,CO4,CO5**
4	(a) Implement the lexical analyzer Program to count the number of vowels and consonants in a given string. (b) Implement the lexical analyzer Program to count the number of characters, words, spaces, end of lines in a given input file.	**CO1,CO2,CO3,CO4,CO5**
5	Implement a 'C' program to calculate First and Follow sets of given grammar	**CO1,CO2,CO3,CO4,CO5**
6	Design Predictive parser for the given language.	**CO1,CO2,CO3,CO4,CO5**
7	Implementation of Shift Reduce Parsing Algorithm.	**CO1,CO2,CO3,CO4,CO5**
8	Design LALR bottom up parser for the given language. (Implementation of calculator using YACC)	**CO1,CO2,CO3,CO4,CO5**
9	Convert the BNF rules into YACC form and write code to generate abstract syntax tree.	**CO1,CO2,CO3,CO4,CO5**
10	Generation of Code for a given Intermediate Code.	**CO1,CO2,CO3,CO4,CO5**

Learning Resources

Text Books

1. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ulman, Second Edition, Pearson Education.
2. Modern Compiler Implementation in C- Andrew N. Appel, Cambridge University Press.

References

1. lex&yacc – John R. Levine, Tony Mason, Doug Brown, O‟reilly
2. Modern Compiler Design- Dick Grune, Henry E. Bal, Cariel T. H. Jacobs, Wileydreamtech.
3. Engineering a Compiler-Cooper & Linda, Elsevier.
4. Compiler Construction, Louden, Thomson.
5. Principles of compiler design, V. Raghavan, Second edition,TMH,2011.

e-Resources and other Digital Material

1. http://www.nptel.iitm.ac.in/downloads/106108052/