

303105350 - Compiler Design Laboratory

Course: BTech

Semester: 6

Prerequisite: Algorithms, Data Structures, Assembly Language Program, Theory of Computation, C/C++ Programming Skills | 203105205 - Data Structure and Algorithms

Rationale : Compiler Design is a fundamental subject of Computer Engineering. Compiler design principles provide an in-depth view of translation, optimization and compilation of the entire source program. It also focuses on various designs of compiler and structuring of various phases of compiler. It is inevitable to grasp the knowledge of various types of grammar, lexical analysis, yacc, FSM (Finite State Machines) and correlative concepts of languages.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, T - Theory, P - Practical

Course Outcome

After Learning the Course the students shall be able to:

After learning the course, the students should be able to

1. Understand the basic concepts; ability to apply automata theory and knowledge on formal languages.
2. Ability to identify and select suitable parsing strategies for a compiler for various cases. Knowledge in alternative methods (top-down or bottom-up, etc.).
3. Understand backend of the compiler: intermediate code, Code Optimization Techniques and Error Recovery mechanisms
4. Understand issues of run time environments and scheduling for instruction level parallelism

List of Practical

1.	Program to implement Lexical Analyzer.
2.	Program to count digits, vowels and symbols in C.
3.	Program to check validation of User Name and Password in C.
4.	Program to implement Predictive Parsing LL (1) in C.
5.	Program to implement Recursive Descent Parsing in C.
6.	Program to implement Operator Precedence Parsing in C.
7.	Program to implement LALR Parsing in C.
8.	To Study about Lexical Analyzer Generator (LEX) and Flex (Fast LexicalAnalyzer)

9.	Implement following programs using Lex. a. Create a Lexer to take input from text file and count no of characters, no. of lines & no. of words. b. Write a Lex program to count number of vowels and consonants in a given input string.
10.	Implement following programs using Lex. a. Write a Lex program to print out all numbers from the given file. b. Write a Lex program to printout all HTML tags in file. c. Write a Lex program which adds line numbers to the given file and display the same onto the standard output.