Department of Computer Science & Engineering

National Institute of Technology Srinagar

Course Title	Compiler Design Lab	**Semester**	7^{th}	
Department	Computer Science & Engineering	**Course Code**	CSL416	
Credits	01	**L**	**T**	**P**
Course Type	Lab	0	0	2

Course Objectives

- To understand the implementation of lexical analyser, parser and other compiler design aspects.
- To write codes for various top-down and bottom-up parsers and verify them for correctness.
- To understand Linux Utility Lex and Yacc tools.

Learning Outcomes

After completing this course the students should be able to understand the compiler coding and working in detail.

Course Synopsis

The Lab intends to make students implement lexical analysers and code for each of the following phases of a compiler:

- Syntax Analysis
- Semantic Analysis
- Intermediate Code Generation
- Code Optimization
- Code Generation

Course Outline / Content

Unit	Topics	Week
1.	Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines.	2
2.	Simulate First and Follow of a Grammar	2
3.	Develop an operator precedence parser for a given language.	1

4.	Construct a recursive descent parser for an expression.	1
5.	Construct a LL(1) parser for an expression	2
6.	Design predictive parser for the given language	1
7.	Implementation of shift reduce parsing algorithm.	1
8.	Design a LALR bottom up parser for the given language.	1
9.	Implement the lexical analyzer using JLex, flex or lex or other lexical analyzer generating tools	1
Text Books		
1.	A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools , Addison-Wesley, 1988.	
2.	C. Fischer and R. LeBlanc. Crafting a Compiler, Benjamin Cummings, 1991.	
References		
1.	A. C. Holub. Compiler Design in C, Prentice-Hall Inc., 1993. Appel. Modern Compiler Implementation in C: Basic Design, Cambridge Press.	
2.	Fraser and Hanson. A Retargetable C Compiler: Design and Implementation, Addison-Wesley.	