T. Y. B. Tech (Computer Science and Engineering) Sem – VI

1. Compiler Construction (PCC - CS601)

TEACHING SCHEME	EXAMINATION SCHEME
Theory : 3 Hrs./Week	**Theory :** ESE 70 Marks CIE 30 Marks
Tutorial : -----	**Term work :** 25 Marks
Practical : 2 Hrs./Week	**Oral : -----**

Pre-requisites: C, System Programming, Data structures

Course Objectives

1. To introduce the fundamentals of compilers and their phases.
2. To design and implement phases of a compiler.
3. To expose the students to various tools like LEX and YACC.

Course Outcomes

1. Recall the compiler phases and compiler construction tools like LEX and YACC.
2. To design and implement Lexical Analyser for a simple language.
3. To design and implement Syntax analyser for a simple expression.
4. To apply Syntax directed translations and Syntax Directed definitions to generate intermediate code.
5. To identify appropriate code optimizing transformation for the given code.
6. To **e**xplain concept of code generation.

UNIT NO.	UNIT NAME & DETAILS	NO. OF LECTURES
1.	**Introduction:** Compilers, Phases of a compiler, Compiler construction tools, cousins of the compiler.	**5**
2.	**Lexical Analysis:** Role of a Lexical analyzer, input buffering, specification and recognition of tokens, finite automata implications, designing a lexical analyzer generator.	**6**
3.	**Syntax Analysis**: Role of Parser, Writing grammars for context free environments, Top-down parsing, Recursive descent and predictive parsers (LL), Bottom-Up parsing, Operator precedence parsing, LR, SLR and LALR parsers	**7**
4.	**Syntax Directed Translation and Intermediate Code Generation**: Syntax directed definitions, construction of syntax tree, S-attributed definitions, L-attributed definitions, Intermediate languages, assignment statements, back patching.	**7**

5.	**Code Optimization**: Principle sources of optimization, optimization of Basic Blocks, loops in flow graphs, Peephole optimization	**5**
6.	**UNIT 6- Code Generation**: Issues in design of a code generator and target machine, Run time storage management, Basic blocks and flow graphs, Next use information and simple code generator, Issues of register allocation, code generation from Dags.	**6**

Term Work

Minimum of 10 to 12 experiments should be carried out based on the following experiments based on following list.

Practical List

Minimum of 10 to 12 experiments should be carried out based on the following experiments.

1. Design of preprocessor for C program.
2. Design a complete lexical analyzer for C language.
3. Program to create a symbol table generator.
4. Design a syntax analyzer for simple expression in c language using top down parsing.
5. Program to create a syntax tree for simple expression in c language using recursive descent parsing.
6. To implement intermediate code generator for Boolean expression in three address code format.
7. Implement intermediate code generator for the conditional statements in three address code format.
8. Implement any one bottom up parsing [LR, SLR, LALR, Operator precedence] technique.
9. To implement a program for code generator from labeled tree.

Text Books

Sr. No.	Title	Author(s) Name	Publication & Edition	Units Covered
1.	Compilers - Principles, Techniques and Tools	A. V. Aho, R .Shethi and J. D. Ullman	Pearson Education	ALL Units

Reference Books

Sr. No.	Title	Author(s) Name	Publication & Edition	Units Covered
1.	Compiler construction	D.M. Dhamdare	Mc-Millan	-
2.	LEX & YACC	Dong Brown, John Levine, Tony Mason	O'Reilly 2nd Edition	Refer for Practical's