COURSE TITLE	**COMPILER DESIGN**
COURSE CODE	**01CT0724**
COURSE CREDITS	**4**

Objective:

1 The purpose of this course is to teach the students about the basic techniques, theory and tools underlie the practice and act of Compiler Construction. This Course introduces the major concept areas of language translation and compiler design.

Course Outcomes: After completion of this course, student will be able to:

1 Understand compiler and different phases. Using this translate program from source code to executable code and files.

2 Explain lexical analysis phase and their connection to language definition through regular expressions and grammars.

3 Explain the syntax analysis phase and differentiate among various parsing techniques and grammar transformation techniques.

4 Use formal attributed grammars for specifying the syntax and semantics of programming languages.

5 Identify the effectiveness of optimization and differences between machine dependent and independent translation.

6 Use the powerful compiler generation tools such as Lex and YACC.

Pre-requisite of course: Basic syntax and semantics of programming languages like object-oriented programming, Data Structure and Theory of Computation

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	2	50	30	20	25	25

Contents : Unit	Topics	Contact Hours
1	**Introduction to Compiler** Review of Languages & Grammar, Translators-Compilation and Compiler, Interpreter and Assembler, overview of linker and loader -Language processors, The Phases of Compiler-Errors Encountered in Different Phases, The Grouping of Phases-Compiler Construction Tools , Programming Language basics, pass structure.	8

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
2	**Scanner** Need and Role of Lexical Analyzer-Lexical Errors, Expressing Tokens by Regular Expressions-Recognition of Tokens, A Language for Specifying Lexical Analyzer, Finite Automata from Regular Expression, Converting Regular Expression to DFA, Minimization of DFA-Language for Specifying Lexical Analyzers, LEX- Design of Lexical Analyzer for a sample Language.	8
3	**Parsing** Top-down Parsing, Predictive parsing, non recursive predictive parsing, First and Follow set, LL(1) grammar, error handling for LL (1), Bottomup parsing, handle pruning, shift reduce parsing, operator precedence parser, LR(0) parser, SLR(1) Parser, Canonical LR(1) Parser, LALR(1) Parser, error detection and recovery in LR Parser, Parser generators (Yacc & Lex)	8
4	**Intermediate Code Generation** Introduction, Intermediate Languages, Types of intermediate forms, Three address Codes: Quadruple & Triples, Translation of Assignment Statements, Boolean expressions, Control Statements, Postfix Translation, Syntax Directed Translation Attributes and Mechanism, Directed Acyclic Graph, Static Single Assignment	6
5	**Memory Management** Introduction, Importance of Memory Management, organization for storage purpose, static allocation, stack allocation, dynamic allocation, different methods of parameter passing, activation record, symbol table	6
6	**Code Optimization** Introduction of Code Optimization, Advantage of code optimization, Types of Code Optimization, Block and Loop Optimization, Global Data Flow Analysis	6
Total Hours		**42**

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Experiment 1** WAP to remove Left Recursion from the grammar.	2
2	**Experiment 2** WAP to remove Left Factoring from the grammar	2
3	**Experiment 3** WAP to verify that the given input is valid identifier or keyword.	2
4	**Experiment 4** WAP to compute FIRST and FOLLOW Set of the given grammar.	2
5	**Experiment 5** WAP to implement Operator precedence parser.	2

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
6	**Experiment 6** Prepare report for Lex, Flex and Yet Another Compiler Compiler Tool.	2
7	**Experiment 7** WAP with the help of Lex and Yacc file to implement Calculator which performs basic operations like addition, subtraction, multiplication and division.	2
8	**Experiment 8** WAP Lex Program to count words, characters, lines, Vowels and consonants from given input.	2
9	**Experiment 9** WAP Lex Program to generate string which is ending with zeros.	2
10	**Experiment 10** WAP Lex Program to check given string is simple or compound string.	2
11	**Experiment 11** WAP Lex Program to count the total number of printf and scanf statement in given C file. Also convert it into readf and write out respectively to another file.	2
12	**Experiment 12** WAP to check given number is positive negative or zero.	2
13	**Experiment 13** WA Lex Program to print HTML tags of given file.	2
14	**Experiment 14** WA YACC Program to generate Calculator.	2
Total Hours		**28**

Textbook :

1 Aho, Lam, Sethi, and Ullman, Compilers: Principles, Techniques and Tools, Pearson, 2014

2 Systerm Programming, D. M. Dhamdhere, Mc Graw Hill Publication , 2011

References:

1 Modern Compiler Design, Modern Compiler Design, Dick Grune, Henri E. Bal, Jacob, Langendoen, Wiley India Publication, 2012

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery

Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
10.00	20.00	30.00	15.00	15.00	10.00

Instructional Method:

1. The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
2. Practical examination will be conducted at the end of the semester for evaluation of performance of students in laboratory.
3. Students may use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory, etc.
4. The course delivery method will depend upon the requirement of content and need of the students. The teacher in addition to conventional teaching method (Chalk and Talk) may use any of the tools such as demonstration, role play, Quiz, brainstorming, Flipped class, Project based learning, Collaborative learning, MOOCs etc. for effective teaching.

Supplementary Resources:

1. https://nptel.ac.in/courses/106/104/106104123/
2. https://nptel.ac.in/courses/106/108/106108113/
3. https://onlinecourses.nptel.ac.in/noc21_cs07/preview

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)