LANGUAGE PROCESSOR - I

Paper Code	**CEN-607**
Course Credits	**4**
Lectures / week	**3**
Tutorial / week	**1**

Course Description

UNIT – I

Introduction to compilation, The tasks of a compiler, Language processing system, Analysis of the Source Program, Phases and Passes in compilers, cousins of compilers, compiler construction tools.

UNIT- II

Role and position of a Lexical analyzer, Input buffering, tokens , lexemes & pattern, review of Regular Expressions, Finite State Machines, Finite Automata based Pattern Matching. Specification and recognition of tokens, a language for specifying lexical analyzer, Design of lexical analyzer generator.

UNIT- III

Role and position of a Parser, A simple Backtracking parser, Predictive Parsing, A review of Context Free Grammar, Derivation tree, Ambiguity. Parsing approaches.
Top-down Parsing: Left recursive grammars, Left factoring, LL (1) Parsing, LL (1) grammars, error recovery in Top down parsers.

UNIT- IV

Bottom Up Parsing technique, Overview of Operator precedence parser, Shift reduce parsing, Finite automata of LR(0) items and LR (0) parsing, SLR parsing, Canonical LR Parsing, LALR Parsing. Compaction of LR parsing table, Error recovery strategies, Yacc: an LALR(1) Parser generator.

UNIT – V

Syntax Directed Definitions and translations, Attributes and Attribute grammar, construction of syntax trees, bottom up evaluation of S attributed definition, L- attributed definition, Top down translation, Analysis of syntax directed definitions.

References / Text Books:

- Aho, Sethi, Ullmann & Lam "Compilers: Principles, techniques and tools", Pearson Education Asia
- Keith Cooper & Linda Torczon, "Engineering a Compiler", Morgan Kaufmann publication.
- Levine, Mason, and Brown, "Lex & Yacc", O' Reilly publication.
- Vinu V. Das, "Compiler Design using FLEX and YACC" PHI.

Computer Usage / Software Requires:

Tools like, LEX \ FLEX, YACC & Bison.