**Compiler Design Syllabus (ECS304)**

**Module I 8 hours**

**Overview of Compilation:** Introduction, The structure of compiler, phases of the compiler, the science of building a compiler: boot strapping, cross compiler.

**Lexical Analysis:** The role of the lexical analyzer, input buffering, specification of tokens, recognition of tokens, the lexical analyzer generator (LEX/FLEX).

**Module II 8 hours**

**Syntax Analysis (Part-I):** Introduction, context free grammars, top down parsing: Brute force parsing, recursive descent parsing, predictive parsing, error recovery in predictive parsing, bottom up parsing, shift reduce parsing, operator precedence parsing, error recovery in operator precedence parsing.

**Module III 8 hours**

**Syntax Analysis (Part-II):** Introduction to LR Parsing: Simple LR parser, more powerful LR parsers canonical LR and look head LR, using ambiguous grammars, error recovery in LR parsers, parser generator (YACC).

**Module IV 8 hours**

**Syntax Directed Translation:** Syntax directed definitions, evaluation orders for SDD's. Intermediate Code Generation: SDD for syntax tree construction, postfix translation, three address codes, translation of arithmetic expressions, translation of array references; Back patching: Boolean expressions, flow of control statements.

**Run Time Environments:** Storage organization, stack allocation of Space.

**Module V 10 hours**

**Machine Independent Optimization:** The principal sources of optimization, basic blocks, flow graphs, loop optimization, DAG representation of basic block, local optimization, introduction to data flow analysis: Reaching definitions, use definition chains, live variable analysis, available expressions.

**Code Generation:** Issues in the design of a code generator, a simple code generator, register allocation and assignment, peephole optimization.

**Text Book(s)**

1. Alfred.V. Aho, J.D.Ullman and Ravi Sethi, Compilers: Principles, Techniques and Tools, 2/e, Pearson Education, 1986.

**References**

1. Alfred.V. Aho and J.D.Ullman, Principles of Compiler Design, Narosa Publications, 2002

2. K Muneeswaran, Compiler Design,Oxford University Press, 2012.

3. John R. Levine, Tony Mason and Doug Brown, Lex & Yacc, O'reilly, 1992.

4. Cooper Linda, Engineering a Compiler, Elsevier, 2011.