1. Principles of Compiler Design (CS-148) **(3-0-2)**

Overview of Compilation: Introduction, Structure of Compiler, Passes, Phases, Compiler Construction tools. Lexical analysis : Finite state machines, regular expression and their role for Lexical analysis, Specification and recognition of tokens , Implementation of Lexical analyzer using NFA, DFA, Lex, LEX Tool, Symbol Table, Syntax Analysis : Formal grammar and their application to Syntax Analysis, Context Free Grammar, Parse trees, Syntax tree and capabilities of CFG, Top down parsing, Bottom up parsing : Yacc automatic parser generator, Semantic Analysis : Attributes of grammar, Symbol Table, Parse tree Abstract Syntax tree, Syntax directed Translation, Syntax directed translation schemes, Overall design of a semantic analyzer Symbol Tables : Symbol table format, organization, Memory allocation, Runtime stack and heap storage allocation, Intermediate code generation : Intermediate code and data structures, Design Issues, Data structure for Intermediate code, Intermediate code generation for data types , statements and Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Object Code Generation and optimization. Practical- Practical on designing lexical analyzer using NFA, DFA , Implementing Top down predictive parser Practical on LEX and Yacc, Generating Intermediate code for simple statements

References:

1. Aho , Ulman, Sethi, Compiler: Principles, Techniques And Tools, Pearson education, 2006.
2. Kenneth C. Louden , Compiler Construction Principles and Practice, PWS Publishing Company, 1997.
3. Thompson Cooper and Linda, Engineering a Compiler-, Elsevier, 2011.
4. John R. Levine, Tony Mason, Doug Brown, lex andyacc , O'reilly, 1992.