# BCA-20
# System Programming

**Unit I : Language Processors**
Introduction to language processor, Interpreters, language processing activities, Program execution, Fundamentals of language processing, phase and passes of language processor, Compiler: Front end, Lexical analysis, Syntax analysis, semantic analysis, memory allocation, code generation, symbol table, Fundamentals of Language Specification : Programming language grammer, Terminal symbols, alphabet and strings, production, derivations, reduction and parse tree, classification of grammar : Type 0, Type 1, Type 2 & Type 3 Grammar, Ambiguity in Grammatic specification, Eliminating ambiguity, Language processor development tools : LEX, YACC

**Unit II : Scanning & Parsing**
Introduction to Scanning, Finite state automaton, Regular expression, Parsing, parse tree, Top Down parsing, predications and backtracking, Top Down parsing without backtracking, Recursive descent parser, LL(1) parser, Bottom up Parsing, Simple precedence grammar, Operator precedence grammars, LALR parsing,

**Unit III: Assemblers**
Elements of Assembly language programming, statement format, Assembly Language statements, Advantage of Assembly Language, Design specification of an assembler, Pass structure of Assembler, Design of A two pass assembler, Pass I of the Assembler, Intermediate code forms, Mnemonic field, Pass II of the Assembler, Architecture of Intel 8088, Design of Assembler

**Unit IV : Compilers and Interpreters**
Aspects of Compilation, Data type, Memory allocation, static and dynamic memory allocation, Accessing nonlocal variables, symbol table requirements, recursion, Compilation of expressions, Intermediate code for expression : Postfix strings, Triples and quadruples, expression tree, compilation of control structure, function and procedure call, Parameter passing mechanisms, code optimization, elimination of common subexpressions, Frequency reduction, Local and global optimization, Interpreters.

**Unit V : Linkers**
Tranlated, linked and load address, Relocation and linking concept, Object module, Design of linker, Linking for Overlays,

## Suggested Readings:


1. Robert Britton: *MIPS Assembly Language Programming*. Prentice Hall, 2003.
2. Compilers principles, techniques & Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman