UNIT I

Chapter 1 : Introduction to Systems Programming and Assemblers 1-1 to 1-44

Syllabus :

- Introduction : Need of System Software, Components of System Software, Language Processing Activities, Fundamentals of Language Processing.
- Assemblers : Elements of Assembly Language Programming, A simple Assembly Scheme, Pass structure of Assemblers, Design of Two Pass Assembler, Single pass assembler.

UNIT II

Chapter 2 : Macro Processors, Loaders and Linkers 2-1 to 2-48

Syllabus :

- Macro Processor: Macro Definition and call, Macro Expansion, Nested Macro Calls and definition, Advanced Macro Facilities, Design of two-pass Macro Processor.
- Loaders : Loader Schemes, Compile and Go, General Loader Scheme, Absolute Loader Scheme, Subroutine Linkages, Relocation and linking concepts, Self-relocating programs, Relocating Loaders, Direct Linking Loaders, Overlay Structure.

UNIT III

Syllabus :

- Phase structure of Compiler and entire compilation process.
- Lexical Analyzer : The Role of the Lexical Analyzer, Input Buffering. Specification of Tokens, Recognition of Tokens, Design of Lexical Analyzer using Uniform Symbol Table, Lexical Errors.
- LEX : LEX Specification, Generation of Lexical Analyzer by LEX.

UNIT IV

Syllabus :

- Role of parsers, Classification of Parsers: Top down parsers-recursive descent parser and predictive parser.
- Bottom up Parsers - Shift Reduce: SLR, CLR and LALR parsers. Error Detection and Recovery in Parser. YACC specification and Automatic construction of Parser (YACC).

UNIT V

Syllabus :

- Need, Syntax Directed Translation, Syntax Directed Definitions, Translation of assignment Statements, iterative statements, Boolean expressions, conditional statements, Type Checking and Type conversion.
- Intermediate Code Formats : Postfix notation, Parse and syntax tress, Three address code, quadruples and triples. Storage Allocation : Storage organization and allocation strategies.

UNIT VI

Syllabus :

- Code Generation : Code generation Issues. Basic blocks and flow graphs, A Simple Code Generator.
- Code Optimization : Machine Independent: Peephole optimizations: Common Sub-expression elimination, Removing of loop invariants, Induction variables and Reduction in strengths, use of machine idioms, Dynamic Programming Code Generation.
- Machine dependent Issues : Assignment and use of registers, Rearrangement of Quadruples for code, optimization.