- Introduction to Compiler
- Analysis and synthesis model of compilation
- Bootstrapping and Porting
- Lexical Analyzer: Input Buffering
- Storage Allocation Strategies
- Type Checking
- LEX
- Dead code elimination
- Loops in flow graphs
- Register allocation and assignment
- Data structure in CD
- Analysis synthesis model of compilation
- Type checking
- Run time environment
- Parameter passing
- Storage organization
- Equivalence of expression in type checking
- Storage allocation strategies
- Function and operator overloading
- Recursive descent parser
- Analysis of syntax directed definition
- Operator precedence parsing
- L-attribute definition
- Syntax analysis CFGs
- Data flow analysis of structure flow graph (SFG)
- Global data flow analysis
- Loop optimization

- Sources of optimization of basic blocks
- Code optimization
- Declaration and assignment in intermediate code generation
- Boolean expression
- Code generation issue in design of code generator

Compiler Design PYQs

- What are the types of passes in compiler ?
- Discuss the role of compiler writing tools. Describe various compiler writing tools.
- What do you mean by regular expression ? Write the formal recursive definition of a regular expression.
- How does finite automata useful for lexical analysis ?
- Explain the implementation of lexical analyzer.
- Write short notes on lexical analyzer generator.
- Explain the automatic generation of lexical analyzer.
- Explain the term token, lexeme and pattern.
- What are the various LEX actions that are used in LEX programming ?
- Describe grammar.
- Explain formal grammar and its application to syntax analyzer.
- Define parse tree. What are the conditions for constructing a parse tree from a CFG ?
- Describe the capabilities of CFG.
- What is parser ? Write the role of parser. What are the most popular parsing techniques ? OR Explain about basic parsing techniques. What is top-down parsing ? Explain in detail.
- What are the common conflicts that can be encountered in shift-reduce parser ?
- Differentiate between top-down and bottom-up parser.Under which conditions

predictive parsing can be constructed for a grammar ?

- Differentiate between recursive descent parsing and predictive parsing.
- What is the difference between S-attributed and L-attributed definitions ?
- What is intermediate code generation and discuss benefits of intermediate code ?
- Define parse tree. What are the conditions for constructing a parse tree from a CFG ?
- Describe the capabilities of CFG.
- Describe the capabilities of CFG.
- What is parser ? Write the role of parser. What are the most popular parsing techniques ? OR Explain about basic parsing techniques. What is top-down parsing ? Explain in detail.
- What are the common conflicts that can be encountered in shift-reduce parser ?
- Differentiate between top-down and bottom-up parser.Under which conditions predictive parsing can be constructed for a grammar ?
- Differentiate between recursive descent parsing and predictive parsing.
- What is the difference between S-attributed and L-attributed definitions ?
- What is intermediate code generation and discuss benefits of intermediate code ?
- Define parse tree. Why parse tree construction is only possible for CFG ?
- Discuss symbol table with its capabilities ?
- What are the symbol table requirements ? What are the demerits in the uniform structure of symbol table ?

Compiler Design Hindi Videos

- 01 Introduction to compiler | passes | classification of compiler in Hindi video
- 02 Compiler vs Interpreter in Hindi video | Compiler Design |
- 03 Analysis and synthesis model of compilation in Hindi video | Compiler Design
- 04 Front end and back end of the compiler in Hindi video| Analysis, Synthesis, Tokens,

Syntax, Semantic

- 05 Phases of compiler in Hindi video | Tokens, Syntax , Semantic Tree, Intermediate code, Assembly code
- 06 Lexical analysis in Compiler Design in Hindi video | Tokens NonTokens
- 07 Semantic analysis phase of compiler in Hindi video | Semantic tree | Symbol table | int to real
- 08 Back end of the compiler in Hindi video | Memory allocation and code generation | Symbol table
- 09 Compiler Design MCQ for GATE, NET, ISRO, KVS, NVS, PGT, DSSSB in Hindi video
- 10 Dead code elimination in compiler design in Hindi video
- 11 Code optimization in Compiler Design in Hindi video
- 12 Symbol Table in Compiler Design in Hindi video
- 13 Bootstrapping the implementation of compilers in Hindi video
- 14 Input Buffering in Compiler Design in Hindi video | 1 and 2 Buffer Scheme
- 15 Boolean algebra in Hindi video | Algebraic functions | Laws of Boolean algebra