UNIT-I

1. Give a brief description about LEX lexical analyzer generator.
2. Explain the pass & phases of translation & interpretation.
3. Draw a block diagram of phases of a compiler and indicate the main functions of each phase.
4. Explain the different tools required for compilation.
5. Explain lexical analysis in detail?
6. Explain syntax analysis phase in detail.

UNIT-II

1. Explain the following terms
   a) Tokens b) lexemes c) patterns
2. Draw the transition diagram for recognizing keywords and identifiers.
3. Explain the role of lexer in compiler.

UNIT-III

1. Write short notes on context free grammars.
2. Convert the following CFG into a suitable form for LL(1) parser. Construct LL(1) parsing table for the converted CFG:
   E → E+E | E-E | E*E | E/E | (E) | id.
3. Are the following grammars LL(1) ? Design top-down parsing procedures for each of the following.
   (a) S → abSa | aaAb | b
       A → ba Ab | b
   (b) S → aSA | ε
       A → bB | cc
       B → bd | ε
4. Explain about predictive parsing.

UNIT- IV

1. Show that the following grammar is LR(1)
   S → AB | bAc | Bc
   A → d
   B → d
2. Explain LALR(1) parser.
   Design an LALR(1) parser for the following grammar:
Program → begin | dec-list | stat-list | end  
Dec-list → dec-list d; | G  
Stat-list → stat-list;stat | stat  
Stat → A | ε  

3. Construct LR(0) parser for the following grammar:  
S → cA | ccB  
A → cA | a  
B → ccB | b  

UNIT-V

1. Explain the LALR parsing table construction algorithm.  
2. Explain Error recovery in LR Parsing  
3. Explain how dangling else ambiguity can be removed.  
4. Explain how to construct LR(1) items with example  

UNIT-VI

1. What is SDT?. Explain how semantic rules can be evaluated.  
2. Explain different storage allocation strategies.  
3. Explain different parameter passing methods.  
4. What is heap? Explain the role of heap management in compilers.  

UNIT-VII

1. Identify and explain any two machine dependent code optimization techniques.  
2. Describe various machine independent code optimization techniques.  
3. Write a brief notes on Local Optimization & Loop optimization.  
4. Explain DAG representation of basic blocks and its Applications.  
5. Explain intermediate code generation techniques.  
6. Explain how the redundant sub expression elimination is performed?  
7. Write about Copy propagation  

UNIT-VIII

1. Write short notes on Object code forms.  
2. What is machine dependent code optimization and how the optimization will be performed?  
3. Explain generic code generation algorithms.  
4. Explain garbage collection via reference counting.