UNIT 1

1. (a) Explain, in detail, lexical analyzer generator.
(b) Describe the lexical errors and various error recovery strategies with suitable examples.
2. (a) Consider the following fragment of 'C' code:
float i, j;
i = i * 70 + j + 2;
Write the output at all phases of the compiler for the above 'C' code.
(b) Write short notes on: input buffering.

3. (a) Explain with one example how LEX program perform lexical analysis for the following patterns in `C': identifier, comments, numerical constants, arithmetic operators.

(b)State the steps to convert a regular expression to NFA. Explain with an example

4. (a) Describe various phases of a compiler? Differentiate a phase and pass? Compare multipass and single pass compiler?

(b) Construct an NFA for regular expression R= (aa/b)*ab convert it into an equivalent DFA?

5.(a) Explain the different phases of a compiler, showing the output of each phase, using the example of the following statement:

position : = initial + rate * 60

(b) Write a LEX program for identifying the keywords and identifiers from the file?

UNIT 2

1. (a) What is top down parsing? Explain in detail.
(b) Consider the following grammar
S -> (L) |a
L -> L, S |S
Construct leftmost derivations and parse trees for the following sentences:
i. (a,(a,a))
ii. (a,((a,a),(a,a))).

2. (a) Consider the following grammar.

S → 0A/1B/0/1

A→0S/1B/1

B → 0A/1S

Construct leftmost derivations and parse trees for the following sentences
i. 0101
ii. 1100101

(b) Consider the following grammar

E → T + E/T

T→ V*T/V

V→id

Write down the procedures for the non-terminals of the grammar to make a recursive descent parser.

3. (a) Draw the syntax tree for the following expression a : = b * - c + b * - c

(b) Differentiate leftmost derivation and rightmost derivation. Show an example for each.

4. (a) Discuss briefly about the classification of parsing techniques.

(b) Write a note on the parse generator '_ YACC.

5. (a) Define left recursion. Is the following grammar left recursive? E → E+E | E*E| a| b

(b) What is an LL(1) parse table? Explain.

UNIT 3

1. (a) What is an SDD? Show an example.
   (b) What are the types of Syntax Directed Translation schemes?

2. (a) Explain the role of intermediate code generator in compilation process.

(b) Define S-attributed SDD and L-attributed SDD.

3. (a) Explain in detail about Implementing L-attributed SDD's.

(b) Write the quadruple, triple, indirect triple for the statement a := b* − c + b* − c.

4. (a) What is a type expression? Explain the equivalence of type expressions with an appropriate examples.

(b) Write a short note on type equivalence and type checking

5. (a) Write a short note on abstract syntax tree.

(b) Distinguish between synthesized & inherited attributes.

UNIT 4

1. (a) What is the use of Symbol table in compilation process? List out various attributes stored in the symbol table.
   (b) Explain different schemes of storing name attribute in symbol table.
2. Explain symbol table organization using hash tables? With an example show the symbol table organization for block structured language.
3. Explain DAG and its use. Write the procedure to construct the DAG for a statement.
4. What are the various operations performed on the symbol table? Explain each of them in detail.
   **14.
5. (a) What is an activation record? Explain how it is related with runtime storage organization?
   (b) Write and explain about heap allocation strategy?
6. (a) Explain the Dynamic storage allocation facilities provided by C language?
   (b) What is dangling reference in storage allocation? Explain with an example

UNIT 5

1. (a) Explain different principal sources of optimization technique with suitable examples.
   (b) Explain machine dependent code optimization with example
2. (a) What is DAG? Construct the DAG for the following basic block
   D := B_C
   E :=A+B
   B := B+C
   A := E-D
   (b) What are the legal evaluation orders and names for the values at the nodes for the DAG of problem (a).
3. (a) Write the importance of global code optimization. Explain redundant sub expression elimination technique across different blocks with example
   (b) Explain with example the various techniques in loop optimization

4. (a) Explain different principal sources of optimization technique with suitable example.

   (b) What is code optimization? What are its advantages?

5. (a) Describe, how redundant expression elimination can be done in loop optimization technique, during global optimization

   (b) Explain about global data flow analysis. List data flow equations for reaching definitionin structureblocks and apply it on the above derived three-address code