

COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours

Max. Marks: 70

PART – A
(Compulsory Question)

- 1 Answer the following: (10 X 02 = 20 Marks)
- (a) Give the phases of a compiler.
 - (b) Define Type 0 Chomsky Hierarchy for Formal Languages.
 - (c) Formally define CFG.
 - (d) Explain in brief the role of Parser.
 - (e) Discuss the types of Intermediate Code.
 - (f) List the five categories of representation of Three address statements.
 - (g) What are the typical places where optimization techniques can be implemented?
 - (h) Illustrate the principal sources of optimization techniques.
 - (i) What are the possible transformations that are applied to peephole optimization?
 - (j) Pick the odd one out:
 - (i) DAG should have directed edges.
 - (ii) Nodes in DAG can have multiple predecessors.
 - (iii) A node in a path in a DAG may repeat.
 - (iv) Nodes in DAG can have multiple successors.

PART – B

(Answer all five units, 5 X 10 = 50 Marks)

UNIT – I

- 2 Explain the different stages of Compiler Design.
- OR**
- 3 Give an algorithm to convert e-NFA to e-free NFA.

UNIT – II

- 4 Describe the steps for LALR Parser.
- OR**
- 5 Describe bottom-up parser.

UNIT – III

- 6 Convert the following expression to Reverse Polish:
 $(1 + 2) * (3 / 4) ^ (5 + 6)$
 Show the stack contents at each step.

OR

- 7 Explain the process of generating three address code.

UNIT – IV

- 8 Describe the process of Dead Code Elimination.
- OR**
- 9 Explain Loop-invariant computations.

UNIT – V

- 10 Describe the various types of machine architectures.
- OR**
- 11 Give the directed acyclic Graph Representation of Basic Blocks.
