

Total No. of Questions: 6

Seat No.:

BACHELOR OF TECHNOLOGY (CBCS - 2023)
B. Tech. Sem-V Computer Science & Engineering AI & ML : WINTER: 2025
SUBJECT: COMPILER CONSTRUCTION

Day : Friday

Time : 02:30 PM-05:30 PM

Date : 12/12/2025

W-30732-2025

Max. Marks : 60

NB :

1. All questions are COMPULSORY.
2. Figures to the right indicate FULL marks.
3. Draw neat labelled diagrams WHEREVER necessary.

- Q. 1 With a neat diagram, explain the phases of a compiler. (10)
OR
- Q. 1 Explain Scanner generator and give lex program specifications with an example (10)
- Q. 2 Compare top-down and bottom-up parsing. Explain how predictive parsing works using FIRST and FOLLOW sets. (10)
OR
- Q. 2 For the grammar below, construct the LL(1) parsing table and test whether it is LL(1): (10)
 $S \rightarrow iEtS \mid iEtSeS \mid a$, $E \rightarrow b$
- Q. 3 Explain the LR(0) parsing technique. Construct canonical LR(0) items for the grammar: (10)
 $S \rightarrow CC$, $C \rightarrow cC \mid d$
OR
- Q. 3 Define syntax-directed translation. Write syntax-directed definitions for infix to postfix conversion and show the annotated parse tree for the expression: $a + b * c$ (10)
- Q. 4 Explain symbol table organization. How are symbols stored, retrieved, and updated during compilation? (10)
OR
- Q. 4 Describe the process of intermediate code generation for procedure calls and parameter passing. (10)
- Q. 5 Explain the principal sources of optimization in a compiler. Discuss how common subexpression elimination and constant folding improve performance. (10)
OR
- Q. 5 Discuss loop optimization techniques in detail. (10)
- Q. 6 Explain the issues in the design of a code generator. Discuss how target machine architecture influences code generation decisions. (10)
OR
- Q. 6 Write an algorithm for register allocation using next-use information with example. (10)
