# PRESIDENCY UNIVERSITY

## BENGALURU

| Make Up Examinations – December 2025 | |
|---|---|
| **Date:** 27-12-2025 | **Time:** 09:30am – 12:30pm |

| School: SOCSE | Program: CSE | |
|---|---|---|
| **Course Code:** CSE3167 | **Course Name:** Compiler Design | |
| **Semester**: MK | **Max Marks**:100 | **Weightage**: 50% |

| CO - Levels | CO1 | CO2 | CO3 | CO4 |
|---|---|---|---|---|
| **Marks** | 26 | 24 | 24 | 26 |

**Instructions:**

*(i) Read all questions carefully and answer accordingly.*

*(ii) Do not write anything on the question paper other than roll number.*

## Part A

**Answer ALL the Questions. Each question carries 2marks.**　　　　　　　**10Q x 2M=20M**

| 1. | Compare Compiler and Interpreter | **2 Marks** | **L2** | **CO1** |
|---|---|---|---|---|
| 2. | Differentiate token, pattern and lexeme | **2 Marks** | **L2** | **CO1** |
| 3. | What are the cousins of a compiler? Name any two. | **2 Marks** | **L1** | **CO1** |
| 4. | What is left factoring in grammar, and how is it applied? Explain with a simple example. | **2 Marks** | **L3** | **CO2** |
| 5. | Eliminate Left Recursion. <br> A-> Ba \| Aa \|c <br><br> B-> Bb \| Ab \|d | **2 Marks** | **L3** | **CO2** |
| 6. | Explain synthesized attributes with example | **2 Marks** | **L1** | **CO3** |
| 7. | List types of Intermediate Code Generation Techniques? | **2 Marks** | **L1** | **CO3** |
| 8. | What is Constant Folding | **2 Marks** | **L1** | **CO4** |
| 9. | What are the rules to choose a Leader? | **2 Marks** | **L1** | **CO4** |
| 10. | Explain any one issue of Code Generation? | **2 Marks** | **L2** | **CO4** |

## Part B

| 11. | a. | Explain Phases of Compiler in detail with the following example<br><br>X=A*B/N+P-D | 10 Marks | L3 | CO1 |
|---|---|---|---|---|---|
| | | **Or** | | | |
| 12. | a. | What is a Lexical Error? List Lexical Errors? | 5 Marks | L2 | CO1 |
| | b. | Explain error recovery in Lexical Analyzer. | 5 Marks | L2 | CO1 |

| 13. | a. | What are the primary reasons for separating lexical analysis from syntax analysis in the compiler design process? Explain briefly. | 5 Marks | L2 | CO1 |
|---|---|---|---|---|---|
| | b. | Construct a transition diagram for recognition of identifiers | 5 Marks | L3 | CO1 |
| | | **Or** | | | |
| 14. | a. | Construct a transition diagram for recognizing relational operators | 5 Marks | L3 | CO1 |
| | b. | Give a regular definition (CGF) for branching statements and conditional expressions. | 5 Marks | L2 | CO1 |

| 15. | a. | Analyze the given context-free grammar (CFG) and Construct Predictive Parsing Table.<br><br>    **E-> TE'**<br>    **E'-> + T E' \| ε**<br>    **T->F T'**<br>    **T' -> * F T' \| ε**<br>    **F-> (E) \|id** | 10 Marks | L3 | CO2 |
|---|---|---|---|---|---|
| | | **Or** | | | |
| 16. | a. | Analyze the step-by-step process of shift and reduce actions to determine whether the string   *id * id + id*    is accepted by the parser.<br><br>    **E → E+E**<br>    **E → E*E**<br>    **E → (E)**<br>    **E → id** | 10 Marks | L3 | CO2 |

| 17. | a. | What is Recursive Descent Parser? Explain with an example | 5 Marks | L2 | CO2 |
|---|---|---|---|---|---|
| | b. | Explain the four possible actions of shift reduce parser. | 5 Marks | L2 | CO2 |

| | | **Or** | | | |
|---|---|---|---|---|---|
| 18. | a. | Construct the SLR Parsing Table for the following Grammar<br><br>    S -> (L)<br>    S -> a<br>    L -> L,S<br>    L -> S | **10 Marks** | **L3** | **CO2** |

| | | | | | |
|---|---|---|---|---|---|
| 19. | a. | Construct an Annotated Parse Tree for (1 + 3) * (4 + 5 ) + 6 | **10 Marks** | **L3** | **CO3** |
| | | **Or** | | | |
| 20. | a. | Define DAG? Construct DAG for the following Example<br><br>**a + a * (b - c) + (b - c) * d** | **10 Marks** | **L3** | **CO3** |

| | | | | | |
|---|---|---|---|---|---|
| 21. | a. | Differentiate S-Attributed and L Attributed SDT. | **5 Marks** | **L2** | **CO3** |
| | b. | Derive Semantic rules for the production.<br><br>$L \rightarrow E$ **n**<br>$E \rightarrow E_1 + T$<br>$E \rightarrow T$<br>$T \rightarrow T_1 * F$<br>$T \rightarrow F$<br>$F \rightarrow (E)$<br>$F \rightarrow$ **digit** | **5 Marks** | **L3** | **CO3** |
| | | **Or** | | | |
| 22. | a. | Construct Three-address code, quadruples, triples and indirect triples for the given statement.<br><br>    **a:=b\*-c+b\*-c** | **10 Marks** | **L3** | **CO3** |

| | | | | | |
|---|---|---|---|---|---|
| 23. | a. | Define Basic Block. Construct the Basic Blocks for the following statements<br><br>**I=1**<br><br>**While(I<10)**<br><br>**{**<br><br>**a=b[i]+c;**<br><br>**i++;**<br><br>**}**<br><br>**Result=a+c** | **10 Marks** | **L3** | **CO4** |
| | | **Or** | | | |
| 24. | a. | List and explain any four key characteristics of peephole optimization with examples. | **5 Marks** | **L2** | **CO4** |

| | b. | State and explain algorithm for Basic Block Construction. | 5 Marks | L2 | CO4 |
|---|---|---|---|---|---|

| 25. | a. | List and briefly explain any four major issues involved in the design of a code generator | 5 Marks | L2 | CO4 |
|---|---|---|---|---|---|
| | b. | Identify the leaders in the following code and justify your selection based on leader rules.<br><br>1: a = b + c<br><br>2: if a > 0 goto 5<br><br>3: d = a - 1<br><br>4: goto 6<br><br>5: d = a + 1<br><br>6: e = d * 2 | 5 Marks | L3 | CO4 |

**Or**

| 26. | a. | Construct a flow graph for the following three address statements<br><br>L1: I = 1<br><br>L2: if (I < 10) then goto TRUE<br><br>L3: goto EXIT<br><br>L4: T1 = b[i]<br><br>T2 = T1 + c<br><br>a = T2<br><br>T3 = I + 1<br><br>I = T3<br><br>goto WSTART<br><br>L5: T4 = a + c<br><br>Result = T4 | 5 Marks | L3 | CO4 |
|---|---|---|---|---|---|
| | b. | State and explain different Ways to Perform Code Optimization | 5 marks | L2 | CO4 |