# Charotar University of Science and Technology [CHARUSAT]

## Chandubhai S. Patel Institute of Technology [CSPIT]

## Department of Computer Science & Engineering

## Assignment-1 & 2

## Last date for submission:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Subject code | : | CS450 | Semester | : | 7 | Academic Year | : | 2023-24 |
| Subject name | : | Compiler Design | | | | | | |

| Sr. No. | Aim | CO |
|---|---|---|
| 1. | What is a compiler? What is the front-end and back-end of the compiler? | 1 |
| 2. | Write a brief note on input buffering techniques. | 1 |
| 3. | Explain the input, output, and action performed by each phases of compiler with example. | 3 |
| 4. | Explain error recovery strategies. | 2 |
| 5. | What is lexical analysis? Which are the tasks performed by lexical analyzer. | 1 |
| 6. | Give the rule to remove left recursive grammar. And Eliminate left recursion from following grammar.<br>S → Aa \| b<br>A → Ac \| Sd \| f | 4 |
| 7. | What is left factoring in CFG? Perform the Left factoring of following Grammar.<br>S → iEtS / iEtSaS / a<br>E → b | 4 |
| 8. | Define token, lexeme, and pattern. Identify the lexemes that makes up the tokens for the following code<br>const p = 10;<br>if( a < p)<br>{ a++ ;<br>If(a== 5)<br>continue ;<br>} | 1 |
| 9. | Consider the following grammar:<br>S' = S#<br>S -> ABC<br>A -> a\|bbD<br>B -> a\| Ɛ<br>C -> b\| Ɛ<br>D -> c\| Ɛ<br>Construct FIRST and FOLLOW for the grammar | 4 |
| 10. | Explain handle and handle pruning | 2 |
| 11. | Explain the language-dependent and machine-independent phases of | 1 |

| | | |
|---|---|---|
| | a compiler. Also, a List of major functions is done by the compiler. | |
| 12. | What do you mean by left recursion and how is it eliminated? | 4 |
| 13. | What is ambiguous grammar? Show that<br>S -> aS\|Sa\|a<br>is an ambiguous grammar. If ambiguous then remove the ambiguity | 4 |
| 14. | Design FIRST and FOLLOW set for the following grammar and solve it using LL(1) grammar.<br>S→ 1AB \| ε<br>A→1AC \| 0C<br>B→0S<br>C→1 | 4 |
| 15. | Explain the activation tree. | 2 |
| 16. | Explain Syntax error Handling. | 2 |
| 17. | What is a language processor? | 1 |
| 18. | Explain role of the parser. | 3 |
| 19. | Explain non-recursive predictive parsing. | 3 |
| 20. | Construction of Non-recursive Predictive Parsing Table. | 3 |

| | | | |
|---|---|---|---|
| **Prepared By:** | Hemang Thakar & Nirav Bhatt | **Date:** | 14/8/2023 |