|  |  |  |
| --- | --- | --- |
| **P.V.P Siddhartha Institute of Technology(Autonomous)** | **Signature of Invigilator with date:** | **Marks Obtained:** |
| **Department of Computer Science and Engineering** |
| **Course: B. Tech** | **Year: III** | **Semester: II** | **Objective: I** |
| **Regulation:PVP20** | **Maximum Marks:10Marks** | **Session: F. N** |
| **A.Y:2024-25** | **Date:20-01-2025** | **Duration: 20 min** |
| **Subject Name: Compiler Design** |
| **Registered Number:** | **Name:** |
| **Answer all the Questions. Each Question carries ½ Mark 20×½ M =10M** |
| **S. No** | **Question** | **CO** | **Level** | **Answer** |
| 1. | **Which of the following does macro expansion** | **CO1** | **L2** |  |
| a. Preprocessor | b. Linker | c. Assembler | d. Loader |
| 2. | **Which of the following is the first phase of Compiler?** | **CO1** | **L2** |  |
| a. Lexical Analysis | b. Syntax Analysis |
| c. Semantic Analysis | d. Code Generation |
| 3. | **Which of the following phase depends on machine?** | **CO1** | **L2** |  |
| a. Lexical Analysis | b. Syntax Analysis |
| c. Semantic Analysis | d. Code Generation |
| 4. | **Which of the following phase produces parse tree as output?** | **CO1** | **L2** |  |
| a. Lexical Analysis | b. Syntax Analysis |
| c. Coe optimization | d. Code Generation |
|  5. | **LEX tool used to genearate \_\_\_\_\_\_\_\_\_\_\_\_.**  | **CO1** | **L2** |  |
| a. lexicalAnalyzer  | b. SyntaxAnalyzer | c. SemanticAnalyzer | d. Code Analyzer |
|  6. | **Which of the following is a token?** | **CO1** | **L2** |  |
| a. Identifier | b. Keyword | c. Operator | d. All |
| 7. | **The regular expression a+ not generates \_\_\_\_.**  | **CO1** | **L2** |  |
| a. a | b. aa | c. € | d. All |
| 8. | **Which of the following is lexical error?** | **CO1** | **L2** |  |
| a. a1 | b. 1a | c. a | d. 1 |
| 9. | **\_\_\_ is a translator.** | **CO1** | **L2** |  |
| a. Compiler | b. Interpreter | c. Assembler | d. All |
| 10. | **Find number of tokens in the following code: int x= =y;** | **CO1** | **L2** |  |
| a. 6 | b. 5 | c. 4 | d. 3 |
| 11. | **Lexical Analyzer groups the characters into meaningful sequences called \_\_.**  | **CO1** | **L2** |  |
| a. Pass | b. Lexeme | c. LEX | d. Phase. |
| 12. | **The regular expression ab+ not generates \_\_\_\_.** | **CO1** | **L2** |  |
| a. abb | b. a | c. ab | d. abbbb |
| 13. | **A compiler can check** | **CO1** | **L2** |  |
| 1. Logical Error
 | 1. Syntax Error
 |
| 1. Both Logical & Syntax Error
 | 1. Not Logical & Syntax Error
 |
| 14. | **Consider the following grammar S ->Ab A ->+ then FIRST(S) =\_\_\_\_.**  | **CO1** | **L2** |  |
| a. **{+}** | b. {a} | c. {$} | d. {A} |
| 15. | **How many components a context free grammar has?** | **CO1** | **L2** |  |
| a. 2 | b. 3 | c. 4 | d. 5 |
| 16. | **A LEX program has \_\_\_\_\_\_\_\_ number of sections** | **CO1** | **L2** |  |
| a. 1 | b. 3 | c. 2 | d. 4 |
| 17. | **In Parse tree, leaf nodes are called?** | **CO1** | **L2** |  |
| a. Terminals | b. Non terminals | c. Sub terminals | d. Half terminals |
| 18. | **A bottom up parser generates** | **CO1** | **L2** |  |
| a. Right Most Derivation | b. Right Most derivation in reverse |
| c. Leftmost derivation | d. Leftmost derivation in reverse |
| 19. | **Shift reduce parsing belongs to class of** | **CO1** | **L2** |  |
| a. Bottom up parsing | b. Top down parsing |
| c. Recursive parsing | d. Predictive parsing |
| 20. | **The grammar E ->EaS | a is \_\_\_\_\_\_recursive.** | **CO1** | **L2** |  |
| a . Left | b. Right | c. Left, right | d. Not |