

BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI  
(END SEMESTER EXAMINATION)

CLASS: MCA  
BRANCH: MCA

SEMESTER : IV  
SESSION : SP/19

SUBJECT: MCA4001 COMPILER DESIGN

TIME: 3 Hours

FULL MARKS: 60

INSTRUCTIONS:

1. The question paper contains 7 questions each of 12 marks and total 84 marks.
  2. Candidates may attempt any 5 questions maximum of 60 marks.
  3. The missing data, if any, may be assumed suitably.
  4. Before attempting the question paper, be sure that you have got the correct question paper.
  5. Tables/Data hand book/Graph paper etc. to be supplied to the candidates in the examination hall.
- 

- Q.1(a) Differentiate between compiler and interpreter. Explain the need for dividing the compilation process into various phases and discuss each of the phases. [8]
- Q.1(b) What is a pass in a compiler? What is the effect of reducing the number of passes? [4]
- Q.2(a) Define parsing. Explain why do we often prefer ambiguous grammar for designing parser. Discuss briefly the difficulties in Top-down parsing. [6]
- Q.2(b) Design LL(1) parser for the grammar having productions:  $S \rightarrow (S)S \mid \epsilon(\text{null})$ . Check whether the given grammar is LL(1) or not from the constructed LL(1) table. [6]
- Q.3(a) What is *shift-reduce* parser? Define handle. Give a suitable example. Explain the conflicts that may occur during shift-reduce parsing. [6]
- Q.3(b) Differentiate between LL(k) and LR(k) parsers, where  $k$  denotes the number of look-ahead tokens. Can you consider  $k=0$  for LL(k)? If so, how does the parser work? Explain why LR parsing is attractive [6]
- Q.4(a) Define *inherited* and *synthesized* attributes. Give example for each. [5]
- Q.4(b) Write an L-attributed SDD for *type declaration* for list of identifiers (in C-language). Show the *annotated parse tree* for the sentence: float  $x, y, z;$  [7]
- Q.5(a) What is the need of intermediate code? List out the types of Intermediated codes. Differentiate between *concrete syntax tree* and *abstract syntax tree*. [6]
- Q.5(b) Construct DAG and QUADRUPLE representation for :  $a+a*(b-c)+(b-c)*d$ . [6]
- Q.6(a) Explain why code optimization is called optional phase. What happened if we do not optimize code? If we optimize code, what will be the issues associated with the optimization? What are the principle sources of optimization? Explain with examples. [6]
- Q.6(b) What is a basic block? Write an algorithm for partitioning a sequence of TAC statements into basic blocks. [6]
- Q.7(a) Explain error detection and error recovery strategy on Predictive parsing. [7]
- Q.7(b) Explain the dynamic storage allocation strategies in detail. [5]

:::::24/04/2019 M:::::