D3ACS2003 **(2 Pages)** **Name.....**
Reg.No.....

THIRD SEMESTER M.Sc. DEGREE EXAMINATION, NOVEMBER 2021

COMPUTER SCIENCE

FCSS3C13-PRINCIPLES OF COMPILERS

Time: 3 Hours **Maximum Weightage: 30**

Section A: Short answer questions. *All* questions can be answered. Each carries *two* weightage (Ceiling 6 weightage).

1. Construct a transition diagram for relational operators.
2. List out the rules for FIRST and FOLLOW.
3. What is the relevance of input buffering in lexical analysis?
4. Why are quadruples preferred over triples in an optimizing compiler?
5. Describe the various fields in an activation record.
6. What is static allocation strategy? What are its limitations?
7. Compare control flow analysis and data flow analysis.

Section B: Short essay questions. *All* questions can be answered. Each carries *four* weightage (Ceiling 12 weightage).

8. Identify any four issues in the design of a Code Generator.
9. Explain the different methods to perform LR parsing with examples.
10. Differentiate between NFA and DFA with examples.
11. Explain about run time storage management.
12. Explain in brief about Type checking and Type Conversion.
13. Explain how DAG will help in intermediate code generation. Construct a DAG and a three address - code for the expression a +a *(b-c)+(b-c)*d.
14. Consider the grammar
    S → (L) | a
    L → L, S | S
    a) What are the terminals, non-terminals and start symbol in the given grammar?
    b) Find parse tree for the following
    - (a,a)
    - (a, ((a,a),(a,a)))

(P.T.O.)

Section C: Essay questions. *All* questions can be answered. Each carries *six* weightage (Ceiling 12 weightage).

15. What is a Flow Graph? Explain how a given program can be converted in to a Flow graph.
16. Explain:
    a) Role of parser in detail.
    b) Shift reduce parsing with the help of an example.
17. Explain the various types of errors generated during the various phases of compiler. How do we recover from these errors?
18. Discuss in detail:
    a) An overview of Region based analysis in optimization.
    b) Optimization of basic blocks.