Total No. of Questions :8] [*Total No. of Printed Pages :2*

Roll No ....

MCA - 405(C)

MCA. IV Semester

Examination, December 2013

Compiler Design

(Elective - I)

Time : Three Hours

Maximum Marks : 70

Note: Attempt any five questions. All questions carry equal marks.

1. a) Distinguish between compiler and interpreter bringing out clearly the situation in which each of them is more suitable than the other. 7

    b) Discuss the challenges is compiler design. 7

2. a) What is the role of lexical analysis is the design of compiler? Is it possible to design a compiler without a distinct lexical analysis phase. 7

    b) Distinguish between NFA and DFA. Compare their powers as token recognizer. 7

3. a) Consider the grammar. 7

    $E \rightarrow E.i(E)|(E)|E?E:|i$, where {E} is the set of non terminal symbols, E is the start symbol and $\{.?:( )\} \cup \{i,i \text{ is identifier}\}$ is a set of terminals symbols. Give a corresponding LL(1) grammar which generates

the same language as the one above. Show the First and Follow sets for each non terminal symbol and the predictive Parsing table Argue that the grammar is LL(1)

    b) Discuss the role of Parser in compiler design. 7

4. a) Discuss the importance of symbol table in compiler design. How is the symbol table manipulated at various phases of compilation. 7

    b) What is the role of intermediate code generation in overall compiler design. 7

5. a) Discuss the necessity of optimization in compilation. 7

    b) What do you mean by runtime storage allocation? Explain the difference between static and dynamic allocation. 7

6. a) Compare and contrast static and dynamic type checking. Give an example of the situation in which dynamic checking is really helpful. 7

    b) Explain simple code generation techniques. 7

7. a) What is recursive descent parsing? Explain with examples. 7

    b) Discuss the tools of lexical analyser. 7

8. Write short notes on any three. 14

    a) Sources of errors

    b) LEX

    c) One pass compiler

    d) Loop optimization

    e) Ambiguity of parse tree.

******