Name : .....

Sixth Semester B.Tech. Degree Examination, June 2015
(2008 Scheme)
08.601 : COMPILER DESIGN (RF)

Time : 3 Hours Max. Marks : 100

PART – A

Answer **all** questions.

1. Differentiate between compiler and interpreter.
2. Differentiate between NFA and DFA.
3. What are the merits and demerits of using a multipass compiler ?
4. Write a short note on input buffering.
5. Explain handle pruning.
6. Explain shift-reduce parsing.
7. Explain left-factoring elimination, with example.
8. What are the conditions to be satisfied by a grammar for being LL(1) ?
9. Define synthesized and inherited translations.
10. Write a note on global optimization. **(10×4= 40 Marks)**

PART – B

Answer **any one** full question from **each** Module.

Module – I

11. a) Explain the phases of a compiler.

    b) Construct context free grammar for the following languages.

    i) $L_1 = \{a^n b^m c^m d^n \mid n \geq 1 \text{and } m \geq 1\}$

    ii) $L_2 = \{a^n b^n c^m d^m \mid n \geq 1 \text{and } m \geq 1\}$

    OR

12. a) Explain back patching, in detail.

    b) Show that the following grammar is ambiguous if 'stat', 'substat' and 'cond' are non-terminals in the given productions that allow them to derive terminal strings : Stat $\rightarrow$ if cond then substat else stat
    | if cond then stat

    Substat $\rightarrow$ if cond then substat else stat.

Module – II

13. a) Compute LEADING and TRAILING for the following grammar.

    $S \rightarrow a \mid \wedge \mid (T)$

    $T \rightarrow T,S \mid S$

    b) Develop the recursive descent parser for the above grammar (Part (a)).

OR

14. Construct simple LR parsing table for the following grammar :

    $S \rightarrow L = R$

    $S \rightarrow R$

    $L \rightarrow *R$

    $L \rightarrow id$

    $R \rightarrow L$

Module – III

15. a) Write quadruples, triples and indirect triples for the expression :

    $-(a + b) * (c - d) - (a + b + c)$

    b) Write a note on the translation of array references.

OR

16. a) Write a note on three-address code and quadruples, with examples.

    b) Write the syntax directed translation scheme for producing postfix notation of a given arithmetic expression (with + and * operators), and give the parse tree (with translations) for the input a + b * c + d . **(3×20 = 60 Marks)**