Code No. 2399 / N

FACULTY OF ENGINEERING

B.E. 3/4 (IT) II – Semester (New) (Main) Examination, April / May 2013

Subject: Compiler Construction

Time: 3 Hours **Max.Marks: 75**

Note: Answer all questions from Part – A and any five questions from Part – B.

PART – A (25 Marks)

1. List the major data structures used in a compiler. (2)
2. What is the role of a lexical analyzer in a compiler? (2)
3. Write the regular definition for the language of C identifiers. (3)
4. Write the types of errors that occur in programming. (2)
5. Remove the left recursion from $E \rightarrow E+n/n$ (2)
6. Define synthesized and inherited attributes. (3)
7. Draw DAG for the following expression

   $a + a*(b-c) + (b-c)*d$ (3)
8. What is short circuit code? (2)
9. Write the design goals of a garbage collector. (3)
10. List the techniques used for semantic preserving transformation. (3)

PART – B (5x10 = 50 Marks)

11.(a) Explain the translation process with a neat diagram. (5)

(b) Give a brief description about lex. (5)

12.(a) Compute FIRST and FOLLOW sets for the following grammar (5)

$S \rightarrow iEtSS'/a$

$S' \rightarrow eS/E$

$E \rightarrow b.$

(b) Write the algorithm for the construction of predictive parsing table. (5)

13. Explain the process of shift-reduce parsing on the i/p string id_1*id_2 for the following grammar. (10)

$E \rightarrow E + T / T$

$T \rightarrow T*F / F$

$F \rightarrow (E) / id$

14.(a) Translate the assignment statement a[i]= b*c - b*d into quadruples and triples. (5)

(b) Using goto-avoiding translation scheme translate the expression

if(a = = b & & c = = d e = = f) x = 1. (5)

15.(a) Discuss the issues in the design of a code generator. (5)

(b) Generate code for the following sequence assuming x,y,z are in memory locations. (5)

if x < y go to $\angle_1$

z = 0

go to L_2

L_1 : z = 1

16.(a) Explain the data-flow analysis framework. (5)

(b) Explain the design of absolute loader. (5)

17. Write short notes on:
    a) Parser generator YACC (4)
    b) Conflicts in shift-reduce parsing. (3)
    c) Rules for type checking. (3)

****