C 26809 **(Pages : 2)** **Name..............................**

Reg. No..............................

SIXTH SEMESTER B.TECH. (ENGINEERING) DEGREE EXAMINATION MARCH/APRIL 2012

IT/CS/PTCS 09 602—COMPILER DESIGN

(2009 Admissions)

Time : Three Hours Maximum : 70 Marks

Part A

Answer **all** *questions.*

1. What are the two parts of compilation ? Explain briefly.
2. Write the regular definition for the following language :
   "All strings of digits with no repeated digits".
3. What is Left recursion? Write the rules to eliminate left recursion.
4. What is the need for code optimization ?
5. How do you calculate the cost of an instruction ?

(5 × 2 = 10 marks)

Part B

Answer any **four** *questions.*

6. Explain the phases of the compiler in detail. Write down the output of each phase for the expression $a := b + c * 50$.
7. Show that the following grammar is ambiguous

   S —> *a*S*b*S/*b*S*a*S/

   For a sentence belonging to the above grammar :

   (i) Draw the parse tree.

   (ii) Construct the left-most and right-most derivations of "*abab*".
8. Write short notes on:

   (i) Left factoring. (ii) Handle pruning.
9. Describe the process of translating Boolean Expression.
10. Explain the issues in the design of a code generator.
11. What do you mean by Peep-hole optimization ? Explain.

(4 × 5 = 20 marks)

Part C

12. (a) With a neat diagram explain various phases of a compiler. List various compiler writing tools you know ?

Or

(b) Construct a minimal DFA for (*a*/*b*)*(*a*/*b*) and write the algorithm for minimizing a DFA.

Turn over

13. (a) (i) Check whether the following grammar is a LL(1) grammar.

S -> *i*E*t*S | *i*E*t*S*e*S | *a*

E -> *b*

(4 marks)

(ii) Explain the first and follow procedures to construct a predictive parser. (6 marks)

Or

(b) Explain the operator precedence parsing algorithm

14. (a) Explain in detail about Run Time Storage Allocation.

Or

(b) Explain about top-down and bottom - up translation.

15. (a) Write the algorithm for constructing the DAG . Explain with Suitable example.

Or

(b) What are the functions of a code generator? Write a simple code generator algorithm and show how it generates code for the statement A = B**c* + B*C + D. Assume only 2 registers are available. What type of optimization can be done at this stage ?

(4 × 10 = 40 marks)