

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 31162

B.E./B.Tech. DEGREE EXAMINATION, MAY/JUNE 2013

Seventh Semester

Computer Science and Engineering

080230045 — PRINCIPLES OF COMPILER DESIGN

(Regulation 2008)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is a compiler?
2. Give an example of tokens.
3. Give regular expression notation to describe a identifier.
4. Write about the tool for generating lexical analyzer.
5. What is a Parser?
6. What are the two basic types of parsers for CFC?
7. What are the four kinds of intermediate code used in compiler?
8. Distinguish between static allocation and dynamic allocation.
9. Distinguish between DAG and flow graph.
10. What are the problems in code generation?

PART B — (5 × 16 = 80 marks)

11. (a) Illustrate the functions of different phases of a compiler. Draw a block diagram to show the decomposition of a compiler. How is a source code parsing through these phases of a compiler with an example? (16)
Or
(b) Describe in detail about the various compiler construction tools. (16)

12. (a) Explain the role of a lexical analyser and its implementation with a suitable example. (16)

Or

- (b) Mention the difference between DFA and NFA. Construct minimum state DFA for the regular grammar

$$(a|b)^*a(a|b)(a|b). \quad (16)$$

13. (a) (i) Define a regular grammar. Construct a regular grammar which will generate all strings of 0's and 1's having both an odd number of 0's and odd number of 1's. (8)

- (ii) Explain predictive parsing technique with an example. (8)

Or

- (b) Construct a recursive descent parsing for

$$E \rightarrow E + T / T$$

$$T \rightarrow T * F / F$$

$$T \rightarrow id / (E)$$

After making the necessary transformation to the above grammar. (16)

14. (a) (i) Explain the intermediate code generation for case statements and procedure calls. (8)

- (ii) Describe backpatching process. (8)

Or

- (b) Write elaborately about translation scheme for flow control statements and Boolean expression. (16)

15. (a) Illustrate the issues in the design of a code generator and give the code generation algorithm. (16)

Or

- (b) (i) Describe the construction of DAG. (8)

- (ii) Write short notes on Peephole optimization. (8)