Reg. No. : □□□□□□□□□□□□

Question Paper Code : 21313

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

Sixth Semester

Computer Science and Engineering

CS 2352 /CS 62 /10144 CS 602 – PRINCIPLES OF COMPILER DESIGN

(Regulation 2008 / 2010)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Define tokens, Patterns and lexemes.

2. Mention the issues in a lexical analyzer.

3. Eliminate left recursion from the following grammar $A \rightarrow Ac/Aad/bd/\varepsilon$.

4. Give examples for static check.

5. What are the various ways of representing intermediate languages?

6. What is back patching?

7. Define basic blocks and flow graphs.

8. Give any four applications of DAG.

9. What is constant folding?

10. What are the properties of optimizing compilers?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Write in detail about the cousins of the complier. (8)

(ii) Explain in detail about the role of Lexical analyzer with the possible error recovery actions. (8)

Or

(b) (i) What are the phases of the compiler? Explain the phases in detail. Write down the output of each phase for the expression a: = b + c * 50. (10)

(ii) Elaborate specification of tokens. (6)

12. (a) (i) Construct a predictive parser for the following grammar. (10)

$S \rightarrow (L) \mid a$

$L \rightarrow L,S \mid S.$

(ii) List all LR (0) items for the following grammar (6)

$S \rightarrow AS / b$

$A \rightarrow SA / a.$

Or

(b) (i) What are the different storage allocation strategies? Explain. (8)

(ii) Specify a type checker which can handle expressions, statements and functions. (8)

13. (a) (i) What are the various methods of implementing three address statements? Explain with example. (8)

(ii) Explain about back patching with an example. (8)

Or

(b) (i) Write down the translation scheme to generate three address code for the assignment statement. (8)

(ii) How would you generate intermediate code for the flow of control statements? (8)

14. (a) (i) Discuss briefly about simple code generation algorithm. (8)

(ii) For the flow graph shown below, write the three address statements and construct the DAG.

(1) t_1=4 * i

(2) t_2:= a[t_1]

(3) t_3:= 4 * i

(4) t_4 := b[t_3]

(5) t_5:= t_2 * t_4

(6) t_6 := prod + t_5

(7) prod := t_6

(8) t_7 := i + 1

(9) i := t_7

(10) if i≤ 20 goto (1). (8)

Or

(b) (i) Explain the concept of register allocation and assignment. (8)

(ii) Discuss labeling algorithm with an example. (8)

15. (a) Write the principle sources of optimization. (16)

Or

(b) (i) Explain the data-flow analysis of structured programs. (8)

(ii) Write global common subexpression elimination algorithm with example. (8)