Reg. No. : ☐☐☐☐☐☐☐☐☐☐☐☐

Question Paper Code : 65036

5 Year M.Sc. DEGREE EXAMINATION, MAY/JUNE 2013.

Sixth Semester

Computer Technology

XCS 362 — COMPILER DESIGN

(Common to : 5 Year M.Sc. — Information Technology/5 Year M.Sc. — Software Engineering)

(Regulation 2003)

Time : Three hours   Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Define cross compiler.

2. What is Bootstrapping?

3. Write the difference between top down and bottom up parsing.

4. What is an operator grammar?

5. What is register allocation?

6. Draw the structure of hash table.

7. What is three address code?

8. Generate three address code for $(b+c)*(b-c)/b$.

9. Define peep-hole optimization.

10. Construct directed acyclic graph for the expression : $a+b*a-b*c$.

PART B — (5 × 16 = 80 marks)

11. (a) (i) Discuss the various phases of compiler. (10)

    (ii) Discuss about compiler construction tools. (6)

Or

    (b) (i) Construct NFA for $a \cdot (a+b)*$ and convert it into DFA. (10)

    (ii) Write the role of lexical analyzer in detail. (6)

12. (a) (i) Construct top down parsing for the input : $a + (b*c)$ using the CFG : (10)

$$S \to S + A \mid A$$
$$A \to A * B \mid B$$
$$B \to (S) \mid id$$

    (ii) Write the role of parser in compiler design. (6)

Or

    (b) (i) Construct operator precedence parsing for the input : $a * (b+c)$ using the CFG : (10)

$$E \to EAE \mid (E) \mid id$$
$$A \to + \mid - \mid * \mid /$$

    (ii) Discuss about syntax errors in detail. (6)

13. (a) (i) Discuss about storage allocation strategies in detail. (8)

    (ii) Discuss about the construction of symbol table. (8)

Or

    (b) (i) Explain the source language issues in run time environment. (8)

    (ii) Explain the various data structures used to construct symbol table. (8)

14. (a) Explain the intermediate code generation for Boolean and assignment statement. (16)

Or

    (b) (i) Explain the various intermediate code representation. (8)

    (ii) Brief back patching with example context. (8)

15. (a) Discuss the issues in code generation and also write the algorithm for code generation. (16)

Or

(b) (i) Discuss about DAG representation of basic block with example. (8)

(ii) Discuss the various code optimization techniques. (8)

———————————