

LIB
27/11/13 FN

Reg. No. :

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

Question Paper Code : 75553

5 Year M.Sc. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2013.

Sixth Semester

Computer Technology

XCS 362/10677 SWE 42 – COMPILER DESIGN

(Common to 5 Year M.Sc. Information technology/M.Sc. Software Engineering)

(Regulation 2003/2010)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. List the different phases of a compiler.
2. Define a regular a expression.
3. Is the following grammar with the three production rules ambiguous?
 $E \rightarrow E + E$ $E \rightarrow E * E$ and $E \rightarrow id$.
4. Compare top down parsing with bottom up parsing.
5. What information is stored in a symbol table?
6. Show the typical layout of the run-time memory.
7. Give an example of an intermediate language.
8. Draw the syntax tree for the arithmetic expression $2 + 3 * 4$.
9. Give an algorithm that partitions the intermediate code into basic blocks.
10. Why is the next-use information computed?

PART B — (5 × 16 = 80 marks)

11. (a) Show how a typical assignment statement gets transformed as it goes through the various phases of a compiler.

Or

- (b) Define a finite automaton. Draw the NFA for the following regular expression :

$b(ab)^* a$ Formulate the regular expression for an identifier in C.

(4 + 8 + 4)

12. (a) Build the predictive parsing table for the following grammar after transforming it, to make it suitable for top down parsing.

$T \rightarrow 1 * F | F$ and $F \rightarrow id$.

Or

- (b) Build a bottom up parser for the grammar given in Q 12. (a).
13. (a) Explain, in detail, the various storage allocation strategies.

Or

- (b) Discuss in detail, with examples, the various parameter passing mechanisms found in programming languages.
14. (a) Write down the syntax-directed definition to generate three address code for assignment statements. Show the annotated parse tree and the three address code generated for an assignment statement of your choice.

(8 + 4 + 4)

Or

- (b) Write down the translation scheme for generating three address code for Boolean expressions using back patching. Show the annotated parse tree and the three address code generated for a Boolean expression of your choice.
15. (a) Explain the various issues in the design of a code generator.

Or

- (b) Explain with examples, the principle sources of code optimization.