

C 22588

(Pages : 2)

Name:

Reg. No:

**SIXTH SEMESTER B.TECH. (ENGINEERING) DEGREE 2014 SCHEME
EXAMINATION, APRIL 2017**

Computer Science Engineering
CS/IT 14 603—COMPILER DESIGN



Time : Three Hours

Maximum : 100 Marks

Part A

*Answer any eight questions.
Each question carries 5 marks.*

1. Define Lexeme and Token. Identify the tokens and lexemes in the following function :

Function gcd (m, n: integer): integer;

begin

 if n = 0 then gcd := m

 else gcd := gcd (n, m mod n)

end;

2. Write a LEX program to print number of identifiers, numbers, and lines in a file.
3. How do you verify whether a given grammar is LL (1) or not without constructing parsing table and also check the following grammar belongs to LL (1) ?

$S \rightarrow iEtSA / a$

$A \rightarrow eS / \epsilon$

1. $E \rightarrow b$

4. Write a YACC program to evaluate the expression with *, +, - operations.
5. Write a procedure for finding FIRST (X) and FOLLOW(X). Here X is any grammar symbol.
6. Write a note on L-attributed and S-attributed definitions with examples.
7. List the storage allocation strategies. Discuss.
8. Discuss the roll of symbol table in compiler design process.
9. Write the syntax directed definition for assignment statements and also generate the three address code for the same.
10. Discuss the issues in the design of a code generator.

(8 × 5 = 40 marks)

Turn over

Part B

Answer all questions:

11. What are the different phases of a compiler? Explain in detail the process of compilation. Illustrate the output of each phase of compilation for the input

$a = b * c + b * 7.5 + b * c$ where b and c are integers and a is a floating point number.

Or

12. List the compiler construction tools and explain each one in detail.
13. Construct an operator precedence parse table for the following grammar.

$S \rightarrow iEtS$

$S \rightarrow iEtSeS$

$S \rightarrow a$

Or

14. Consider the grammar

$berpr \rightarrow bexpr \text{ or } bterm \mid bterm$

$bterm \rightarrow bterm \text{ and } bfactor \mid bfactor$

$bfactor \rightarrow \text{not } bfactor \mid (bexpr) \mid \text{true} \mid \text{false}$

Construct a predictive parser for the above grammar.

15. Explain in detail top-down and bottom-up translations with examples.

Or

16. What is the activation record in Syntax-directed translation? Also discuss the roll of the activation record in access to non-local names.

17. Write a three address code for the following Boolean expressions with syntax tree.

while $a < b \ \&\& \ c < d$

if $e < f$ then

$a = a + b ;$

else

$a = a - b ;$

Or

18. Discuss the code optimization techniques with examples.

(4 × 15 = 60 marks)