

Reg. No.

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

B.E. / B.TECH. DEGREE EXAMINATIONS, DEC 2019

Sixth Semester

CS16602 – COMPILER DESIGN*(Computer Science and Engineering)***(Regulation 2016)****Time: Three Hours****Maximum : 100 Marks**Answer **ALL** questions**PART A - (10 X 2 = 20 Marks)**

	CO	RBT
1. Differentiate Compiler and Interpreter.	1	AN
2. Give the general structure of language processing system.	1	U
3. Define Token, Lexeme and Pattern.	2	R
4. What is meant by handle pruning?	2	R
5. Define 'Dangling-Else' ambiguity.	3	R
6. How to eliminate left recursion in a grammar?	3	AN
7. Construct the abstract syntax tree for the expression $a=b*-c + b*-c$	4	AP
8. Write a note on L-attributed and S-attributed definitions with examples.	4	U
9. List the issues in the design of a code generator.	5	R
10. What is activation record?	5	R

PART B - (5 X16 = 80 Marks)

11. (a) (i) Explain in detail about Phases of Compiler with neat diagram. (16) 1 AN
Write the output of each phase for the following statement.
count = count * times + 1
- (OR)**
- (b) (i) List & Explain in detail about various compiler construction tools. (8) 1 AP
- (ii) Explain in detail about grouping of compiler phases. (8) 1 U
12. (a) Explain in detail about specification of tokens and write how to express the regular expression for given tokens. (16) 2 AP

(OR)

- (b) Explain LEX tool with their specification. Also write the LEX program to count the number of characters, words, lines, vowels and consonants in the given input file. **(16) 2 AN**

13. (a) Construct SLR parsing table for the grammar: **(16) 3 AP**
 $E \rightarrow E+E \mid E^*E \mid (E) \mid id$

(OR)

- (b) Show that the following grammar: **(16) 3 AN**
 $S \rightarrow Aa \mid bAc \mid dc \mid bda \quad A \rightarrow d$ is LALR (1) but not SLR (1).

14. (a) (i) Discuss in detail about the Simple Type Checker. **(8) 4 AN**
 (ii) Briefly explain about the source language issues and storage organization for run time environment. **(8) 4 AN**

(OR)

- (b) Explain in detail about the various storage allocation strategies for run time environment. **(16) 4 AP**

15. (a) Explain in detail about the Principle Sources of Code Optimization with example. **(16) 5 AP**

(OR)

- (b) Construct a DAG and a three address-code for the expression : **(16) 5 AN**
 $a + a*(b - c) + (b - c)*d$