Reg. No. | | | | | | | | | | | | |

Question Paper Code : 57263

B.E./ B.Tech. DEGREE EXAMINATION, MAY/JUNE 2016

Sixth Semester

Computer Science and Engineering

CS 6660– COMPILER DESIGN

(Regulations 2013)

Time : Three Hours **Maximum : 100 Marks**

Answer ALL questions.

PART – A (10 × 2 = 20 Marks)

1. What are the two parts of a compilation ? Explain briefly.
2. Illustrate diagrammatically how a language is processed.
3. Write a grammar for branching statements.
4. List the operations on languages.
5. Write the algorithm for FIRST and FOLLOW in parser.
6. Define ambiguous grammar.
7. What is DAG ?
8. When does Dangling references occur ?
9. What are the properties of optimizing compiler ?
10. Write three address code sequence for the assignment statement

    d:= (a-b) + (a-c) + (a-c).

PART – B (5 × 16 = 80 Marks)

11. (a) Describe the various phases of compiler and trace it with the program segment (position:= initial + rate * 60). **(16)**

OR

(b) (i) Explain language processing system with neat diagram. **(8)**

(ii) Explain the need for grouping of phases. **(4)**

(iii) Explain various Error encountered in different phases of compiler. **(4)**

12. (a) (i) Differentiate between lexeme, token and pattern. **(6)**

(ii) What are the issues in lexical analysis ? **(4)**

(iii) Write notes on regular expressions. **(6)**

OR

(b) (i) Write notes on regular expression to NFA. Construct Regular expression to NFA for the sentence (a|b)* a. **(10)**

(ii) Construct DFA to recognize the language (a/b)* ab. **(6)**

13. (a) (i) Construct Sack implementation of shift reduce parsing for the grammar **(8)**

E -> E+E

E -> E*E

E -> (E)

E -> id and the input string id1 + id2 *id3

(ii) Explain LL(1) grammar for the sentence S->iEts | iEtSeS | a E->b. **(8)**

OR

(b) (i) Write an algorithm for Non recursive predictive parsing. **(6)**

(ii) Explain Context free grammers with examples. **(10)**

14. (a) (i) Construct a syntax directed definition for constructing a syntax tree for assignment statements. **(8)**

$S \rightarrow$ **id** : = E

$E \rightarrow E1 + E2$

$E \rightarrow E1 * E2$

$E \rightarrow -E1$

$E \rightarrow (E1)$

$E \rightarrow$ **id**

(ii) Discuss specification of a simple type checker. **(8)**

OR

(b) Discuss different storage allocation strategies. **(16)**

15. (a) Explain Principal sources of optimization with examples. **(16)**

OR

(b) (i) Explain various issues in the design of code generator. **(8)**

(ii) Write note on simple code generator. **(8)**

———————