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

Question Paper Code : 52873

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2019.

Sixth Semester

Computer Science and Engineering

CS 6660 — COMPILER DESIGN

(Common to Information Technology)

(Regulation 2013)

(Also common to PTCS 6660 – Complier Design for B.E. (Part-Time) for Fifth Semester – Computer Science and Engineering – Regulation 2014)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. List the attributes stored in symbol table.

2. Why is compiler optimization essential?

3. Discriminate tokens, patterns and lexemes.

4. Write the regular expression for all valid identifiers.

5. What is meant by handle pruning?

6. Mention the purpose of YACC.

7. What are the various ways of passing a parameter to a function?

8. Write the grammar for flow control statement while-do.

9. Define address descriptor.

10. Write the object code sequence for t:=a+b produced by a typical code generator.

PART B — (5 × 13 = 65 marks)

11. (a) Explain various phases of compiler in detail. Write the output of each phase of the compiler for the expression c :=a + b * 12. (13)

Or

(b) (i) What are the characteristics of compiler construction tools? Explain how compiler construction tools help in implementation of various phases of a compiler. (7)

(ii) Differentiate call-by-value and call-by-reference parameter passing mechanisms with suitable examples. (6)

12. (a) (i) Analyze the role of lexical analyzer with suitable examples. (7)

(ii) Draw and explain the transition diagram that recognizes the lexemes matching the token relop (relational operator). (6)

Or

(b) Write the subset construction algorithm. Using the subset construction algorithm, convert the regular expression (a | b)*abb to DFA. (13)

13. (a) Write the algorithm for construction of LR parsing table for a given grammar. Construct the LR parsing table for the following grammar : (13)

$E \rightarrow E + T$

$E \rightarrow T$

$E \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id$

Or

(b) Write the algorithm for construction of LALR parsing table for a given grammar. Using the algorithm for construction of LALR parsing table construct the LALR parsing table for the following grammar. (13)

$S' \rightarrow S$

$S \rightarrow aAd \mid bBd \mid aBe \mid bAe$

$A \rightarrow c$

$B \rightarrow c$

14. (a) (i) Describe syntax-directed translation schemes with appropriate examples. (7)

(ii) Explain how type conversion is performed with suitable examples. (6)

Or

(b) Explain various techniques for storage allocation with examples. (13)

15. (a) (i) Write and explain the algorithm for construction of basic blocks. (7)

(ii) Construct the DAG for the following basic block. (6)

```
x = a[i]
a[j]= y
z = a[i]
```

Or

(b) Explain the algorithm that generates code for a single basic block with suitable examples. (13)

PART C — (1 × 15 = 15 marks)

16. (a) In SQL, keywords and identifiers are case-insensitive. Write a Lex program that recognizes the keywords SELECT, FROM, and WHERE (in any combination of capital and lower-case letters), and token ID, which may be any sequence of letters and digits, beginning with a letter. (15)

Or

(b) A simple matrix-multiplication program is given below :

```
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        c[i][j] = 0.0;
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        for (k=0; k<n; k++)
            c[i][j] = c[i][j] + a[i][k]*b[k][j];
```

(i) Translate the program into three-address statements. Assume the matrix entries are numbers that require 8 bytes, and that matrices are stored in row-major order. (7)

(ii) Construct the flow graph for the code from 1. (6)

(iii) Identify the loops in the flow graph from 2. (2)

———————