Course Code : CST 358 **QVRU/RS – 22 / 1013**

Sixth Semester B. E. (Computer Science and Engineering) Examination

COMPILER DESIGN

Time : 2 Hours] [Max. Marks : 40

Instructions to Candidates :—

(1) All questions carry marks as indicated against them.
(2) Due credit will be given to neatness and adequate dimensions.
(3) Assume suitable data and illustrate answers with neat sketches wherever necessary.

1. (a) Construct the DFA for the regular expression $a*b*(cb)*$ using the syntax tree method by finding the Nullable, Firstpos and Lastpos. 4 (CO 1)

   (b) Write a regular expression for the following :—

   (i) Negative decimal floating numbers less than 10 and can have a maximum of 2 digits after the decimal point.

   (ii) Integer numbers. Comma is allowed as a separator between groups of numbers to make integer easier to read. Number cannot begin or end with comma. 2 (CO 1)

2. (a) Design the LL(1) parsing table for the following grammar :—

   $S \rightarrow ABa \mid b$
   $A \rightarrow cBCD \mid \varepsilon$
   $B \rightarrow CdA \mid ad$
   $C \rightarrow eC \mid \varepsilon$
   $D \rightarrow bSf \mid a$

   Check whether the given grammar is LL(1) or not ?
   Show the parsing of the string "eeda" and "adb". 8 (CO 2)

OR

   (b) Construct the LR(1) Parsing table for the following grammar :

   $A \rightarrow XYZ$
   $X \rightarrow yY \mid zZ$

$Y \rightarrow aZ$

$Z \rightarrow BA \mid ZXc$

Show string parsing for the string "yzac" and state whether the string is valid or invalid for the given grammar. 8 (CO 2)

3. (a) Generate the Three Address Code (TAC) using SDTS for the given language construct.
Show the annotated parse tree and the TAC generated.

```
while (not (a > b)) do
begin
  if (c > 5) then
      c = c – 1 ;
end
```

Also give the value of target of S.next. 4 (CO 3)

(b) Generate the three – address code for following array reference :

A[I, J + 1] = B[I, C [I, J]] + D[I, J + 1]

Where w = 4 and the size of array A, B, C and D are 10×20, 10×5, 5×5 and 10×5 respectively. 4 (CO 3)

4. (a) Discuss Phrase level recovery in LR parser.
Consider the grammar $E \rightarrow E + E \mid E * E \mid (E) \mid id$. The LR parsing table for the given grammar is —

	id	+	*	S	E
0	S2				1
1		S3	S4	accept	
2		R3	R3	R3	
3	S2				5
4	S2				6
5		R1	S4	R1	
6		R2	R2	R2	

Implement the phrase level error recovery routines for the LR parsing and trace the behavior of parser on input id + id * +id. 5 (CO 1)

(b) Identify the phase of compiler where the following error occurs.

(i) printf("Compiler Design"); #

The error in the statement is : Illegal character # appears at the end.

(ii) int x[50], y ;

---

x = y

The error in the code is : incompatible type of a and b. 1 (CO 1)

5. (a) Consider the given three address code. Generate the program flow graph and state the loops in the program flow. Compute the IN – OUT GEN KILL equation :—

i = m – 1
j = n
a = v1
i = i + 1
j = j – 1
if e1 goto (9)
i = v3
goto (10)
a = v2
If e2 goto (4). 4 (CO 4)

(b) Consider the given Three Address Code, Construct the DAG and perform common subexpression elimination :

t1 = 4 * i
t2 = a [t1]
t3 = 4 * i
t4 = b [t3]
t5 = t2 * t4
t6 = prod + t5
prod = t6
t7 = i + 1
i = t7
if i <= 20 goto 1. 2 (CO 4)

6. (a) Consider the expression $S = (b + c) - ((m + n) - k))$. Generate the target code using getreg( ) function considering single register. 3 (CO 4)

OR

(b) Construct the DAG for the following expression : Apply labelling algorithm to find the number of registers required. Also determine the optimal sequence using heuristic algorithm :

$Z = X - Y + X * Y * U - V / W + X + V.$ 3 (CO 4)

(c) Apply the gencode algorithm to generate the target code for the following three address code :—

(1) $t1 = c / d.$

(2) $t2 = t1 - f.$

(3) $t3 = b + e.$

(4) $t4 = t2 * t3.$ 3 (CO 4)