Mode of Examination: Online
M.Sc. Semester- IV Examination, 2021

2021
Subject: Computer Science
Paper Code & Name: CSM401(04) & Compiler Design (Elective-I)
Full Marks: 70
Date: 26.07.2021 **Time: 3:30 hours** **Duration: 12:00 hrs -15:30 hrs**

Please follow the instructions below carefully.
Promise not to commit any academic dishonesty
Candidates are required to answer in their own words as far as applicable.
Each Page of answer scripts should have your examination Roll Number on the right side top corner of your answer script / Script may be a softcopy also.
The name of the scanned copy of the assessment script will be of the following format:
CSM401-04-CD-Roll-Number.pdf
(Example: CSM401-04-CD-C91-CSC-191001.PDF)
The subject of the mail should be the file name only.
The scanned answer script is to be sent to cucse2020@gmail.com
The report should have an index page.
30 minutes is allotted for scanning and uploading your answer script.
The answer script may not be accepted after the scheduled time.

Answer Question No. **1, 2**, and **any Four** from the rest.
The figures in the margin indicate full marks.

1. Answer ***any*** 5 questions 2 X 5 = 10

(a) Describe the type of grammar with the following set of production rules

$P = \{S \rightarrow a \mid b \mid Sa \mid bS\}$

(b) What is handle pruning in the bottom-up parsing technique?

(c) Differentiate between dead code elimination and redundant code optimization.

(d) What are the problems associated with Top-Down Parsing?

(e) Differentiate between synthesized translation and inherited translation

(f) What are the data structures used in the symbol table?

(g) What is an activation record? Write the various fields of an Activation Record.

2. Answer ***any*** 5 questions 4 X 5 = 20

(a) Design DAG for the expression: (a − b + c) / (b − c) ∗ d − (a − b)

(b) Write three address code for the following program

```
i =1;
t =3;
while ( i <=10)
{
        x [ i ] = t ∗ i ;
        i=i +1;
}
```

(c) Write a short note on the recursive descent parser.

(d) Classify the errors and discuss the errors in each phase of Compiler Design.

(e) What is control and data flow analysis? Explain with an example.

(f) Construct syntax tree and postfix notation for the following expression:

(a+(b*c)^d-e/(f+g))

(g) Write quadruples, triples and indirect triples for the expression:

-(a*b)+(c+d)-(a+b+c+d)

3. (a) State the difference between *LR*(*k*) and *LL*(1) parsing techniques.

(b) How can you check whether a grammar is *LL*(1) grammar or not?

(c) What is shift-reduce conflict and how it is handled in the *LALR* parser generation technique?

(3+3+4)

4. Construct a syntax-directed translation to convert *infix* expression to *postfix* expression. Explain its working with a suitable example. Identify the basic blocks of the following function and then construct the program flow graph.

```
fact ( int  n )
{       int  i , f =1;
        for ( i =1; i < n ; i ++)
        f ∗= i ;
}
```

Why is code optimization used in the compiler design? (3+5+2)

5. What do you mean by the cost of a three address instruction? Find such cost for the following program segment:

```
t1 = a ∗ a
t2 = a + b
t3 = 2 ∗ t2
t4 = t1 + t3
t5 = b ∗ b
t6 = t4 + t5
```

Explain how this cost can be reduced. Now apply code optimization for the above program fragment.

(2+3+2+3)

6. (a) What is an LALR parser? Construct the set of LR(1) items for this grammar:

S →CC
C → aC
C → d

(b) Compare and contrast among SLR Parser, LALR parser, and Canonical LR Parser.

(6+4)

7. (a) Consider the following Grammar:

A→ ABd|Aa|a
B→ Be|b

Remove left recursion.

(b) Perform left factoring in the following grammar:

A→ aAB|aA|a
B→ bB|b

(5+5)

8. (a) What are the properties of the code generation phase? Also, explain the design issues of this phase.

(b) What are basic blocks? Write an algorithm for partitioning into basic blocks.

(5+5)