COMPILER CONSTRUCTION
(CSEN 4101)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i)
```
int  main()
{
    int  xI, Nx;
    for ( xI = 0; xI < Nx; xI++);
}
```

(a) No Compilation Error (b) Only Lexical Error
(c) Only Syntactic Error (d) both lexical and syntactic errors

(ii) Consider the SDT, A → BC {A.i = B.i} is _____
(a) S-attributed (b) L-attributed (c) both (d) none

(iii) What will be the postfix representation for the below infix expression?
x + -y * (-y + z)
(a) xy+-y-z*+ (b) xy-y-z+*+ (c) xy-+y-z+* (d) none of these

(iv) Statement starting from a leader up to the next leader is called _____
(a) trailer (b) basic block (c) DAG (d) Flow graph

(v) Code generator is dependent on
(a) type of input and output (b) register allocation
(c) (a) & (b) both (d) none of these

(vi) {S, A, B} is the non-terminal alphabet and {a, b} is the terminal alphabet of the CFG. S is the start symbol. The set of production rules are given below,
S->aB S->bA
B->b A->a
B->bS A->aS
B->aBB A->bAA
(a) aabbbb (b) abbbba (c) aabbab (d) aaaabb

(vii) Machine independent optimization is applied on the code when it is in ____
(a)Intermediate form (b) Assembly code
(c)Target code (d) Machine code.

(viii) Which one of the following statement is true?
(a) Canonical LR parser is more powerful than LALR parser
(b) SLR parser is more powerful than LALR
(c) LALR parser is more powerful than canonical LR parser
(d) SLR parser, canonical LR parser and LALR parser all have the same power.

(ix) Find the number of tokens in the following code:
if(x>=y) z=0;
(a) 8 (b)9 (c) 10 (d)11.

(x) YACC builds up
(a) SLR Parsing table (b) LALR Parsing table
(c) Canonical LR Parsing table (d) None of the above.

Group – B

2. (a) A lexical analyser uses the following patterns to recognize three tokens T1, T2, and T3 over the input symbols {a, b, c}
T1 : a?(b|c)*a T2 : b?(a|c)*b T3 : c?(b|a)*c
If the string "bbaacabc" is processed by the analyser, which sequence of tokens it outputs?

(b) Construct a DFA that accepts all possible strings over the input symbols {0, 1} that do not contain 011 as substring.

(c) Give the CFG, which generates all positive even integers up to 998.

3 + 6 + 3 = 12

3. (a) Give the output of each phase of compiler for the following source text
while(i < 10)
i = i + 1;

(b) Find the regular grammar equivalent to the regular expression given below:
(ab)*ba(ab)*.

(c) Can you implement top down parsing on the CFG given below:
S → aSSbS | aSaSb | abb | b
If not then what will be an equivalent CFG, where you can implement top down parsing?

5 + 3 + 4 = 12

Group – C

4. (a) Check whether the following grammar is LL(1) or not?
S → aAC | bB
A → Abc | Abd | e

B → f | g
C → h | i

(b) Find the number of conflicts, if any in the SLR(1) parsing for the below mentioned CFG.

S → A
A → AB | ∈ {∈ represents NULL}
B → aB | b

(c)
```
program main;
    procedure P;
        procedure Q;
          begin R; end
        procedure R;
          begin Q; end
    begin R; end
begin P; end
```
Consider the above mentioned nested function definition and show the allocation of activation records along with static and dynamic links

3 + 6 + 3 = 12

5. (a) Construct an LR(1) parsing table for the following grammar:

S → A
A → AB | ∈ {∈ represents NULL}
B → aB | b

(b) Generate annotated parse tree for the string "3+2-4" using
E→ E+T|E-T|T
T→ 0|1|2|3|4|5|6|7|8|9

8 + 4 = 12

Group – D

6. (a) Generate the three address code for the below program fragment:
```
sum = 0;
for( i = 1; i <= 20; i++)
        sum = sum + a[i] + b[i];
```

(b) Construct a DAG for the below expression

p = p + q / (r – s) + (r – s) * q

then generate quadruples, triples, and indirect triples for the same expression.

6 + (1.5 × 4) = 12

7. (a) Explain the following loop optimization techniques with examples:
(i) Frequency reduction
(ii) Strength reduction
(iii) Code motion

(b) What is an activation record? Explain clearly the components of activation record.

(c) Define inherited attribute with suitable examples.

(2 × 3) + 4 + 2 = 12

Group – E

8. (a) Differentiate Phrase level recovery and Panic mode recovery.

(b) Give the definitions of the code optimization techniques given below with suitable examples.
(i) Copy propagation
(ii) Strength reduction

(c) Consider the following code snippet

```
A = 4
t1 = A * B
repeat
{
   t2 = t1/C
   if (t2 ≥ W)
  {
        M = t1 * k
        t3 = M + I
  }
  H = I
  M = t3 - H
} until (T3 ≥ 0)
```

Now draw the control flow graph for it, check if any optimization can be incorporated.

3 + (2.5 × 2) + 4 = 12

9. (a) Translate the following expression into machine code and show the register descriptor and address descriptor while the instructions are generated. Assume that only two registers are available

x = (a + b) – ((c + d) - e)

(b) Write a short note on “Peephole optimization”.

8 + 4 = 12

Department & Section	Submission link:
CSE A	https://classroom.google.com/c/MTIyNDM4MjA5OTg2/a/Mjc0NTUzNjM2NDE2/details
CSE B	https://classroom.google.com/c/MTIyNDA4ODI1NjM3/a/Mjc0NzEzNzYxOTc3/details

CSE C	https://classroom.google.com/w/NTUxNTg2MzQzODVa/tc/Mjc0NzIwNjU1MDc3
BACKLOG	https://classroom.google.com/w/MjgyNTI3NDEyMDY5/tc/MjgyNTI3NDEyMjM4