COMPILER DESIGN
(INFO 3132)

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) Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G has n2 states.
(a) n1 is necessarily less than n2 (b) n1 is necessarily equal to n2
(c) n1 is necessarily greater than n2 (d) None of the mentioned.

(ii) If a state does not know whether it will make a shift operation or reduction for a terminal is called
(a) Shift/reduce conflict (b) Reduce /shift conflict
(c) Shift conflict (d) Reduce conflict.

(iii) Shift reduce parsers are
(a) Top down Parser (b) Bottom Up parser
(c) May be top down or bottom up (d) None of the mentioned.

(iv) Which of the following derivations does a top-down parser use while parsing an input string?
(a) Leftmost derivation (b) Leftmost derivation in reverse
(c) Rightmost derivation (d) Rightmost derivation in reverse.

(v) Type checking is normally done during?
(a) Lexical Analysis (b) Syntax Analysis
(c) Syntax Directed Translation (d) Code generation.

(vi) A system program that combines separately compiled modules of a program into a form suitable for execution is
(a) Assembler (b) Linking Loader
(c) Cross Compiler (d) None of the mentioned.

(vii) Which of these is true about LR parsing?
(a) Is most general non-backtracking shift-reduce parsing
(b) It is still efficient
(c) Both a and b
(d) None of the mentioned

(viii) The graph that shows basic blocks and their successor relationship is called
(a) Dag (b) Flow Graph
(c) Control Graph (d) Hamilton Graph

(ix) The optimization which avoids test at every iteration is
(a) Loop unrolling (b) Loop jamming
(c) Constant folding (d) None of the mentioned.

(x) A grammar that produces more than one parse tree for some sentence is called
(a) Ambiguous (b) Unambiguous
(c) Regular (d) None of the mentioned.

Group – B

2. (a) Write the phases and sub-phases of compiler with its output after completion of each phases and sub-phases.

(b) What is the output of lexical analyzer? Define different types of tokens with example. Identify the tokens in the following statement:
int a, b, c;
a=(b+c);

4 + (2 + 3 + 3) = 12

3. (a) Write Context Free Grammar (CFG) for the regular expression r=0*1(0+1)*

(b) Define Regular Expression(RE) recursively with example. Write a regular Expression(RE) for the set of strings whose 2nd bit from right end is 1 and 4th bit from right end is 0.

6 + (2 + 4) = 12

Group – C

4. (a) Draw block diagram of predictive parser with stack and briefly describe the components. Define Handles with example.

(b) Compute FRIST() and FOLLOW() of the following grammar:
$S \rightarrow iEtSS_1 \mid a$
$S_1 \rightarrow eS \mid \epsilon$
$E \rightarrow b$

(4 + 2) + (3 + 3) = 12

5. (a) Consider the following grammar and test whether the grammar is LL(1) or not. State the reason.

$S \rightarrow 1AB \mid \epsilon$
$S_1 \rightarrow 1AD \mid 0D$
$B \rightarrow 0S$
$D \rightarrow 1$

(b) Define augmented grammar with example. Draw parse tree for the following code fragment:

```
if(basic> 25,000)
salary =basic*1.7;
else
salary= basic*1.8 + PF;
```

6 + (2 + 4) = 12

Group – D

6. (a) What are the different ways to represent intermediate code? Consider the following statements to construct 3-address code with quadruples:

```
x:= y+z+k;
if  x>10 then
x:=y*z*k;
else x:=y+z-k;
```

(b) Define annoted parse tree. Given the Syntax-Directed Definition below with the synthesized attribute val, draw the annotated parse tree for the expression $4 * 5 + 6_n$

$L \rightarrow E_n$	L.val = E.val
$E \rightarrow T$	E.val =T.val
$E \rightarrow E + T$	E.val = E.val + T.val
$T \rightarrow F$	T.val = F.val
$T \rightarrow T * F$	T.val = T.val * F.val
$F \rightarrow (E)$	F.val = E.val
$F \rightarrow digit$	F.val = digit.lexval

(1 + 5) + 6 = 12

7. (a) Write short note on activation record.

(b) Explain code motion with a example.

6 + 6 = 12

Group – E

8. (a) What is DAG? Write the steps for DAG construction. Draw DAG for the following expressions:

```
m = n+o
n = n - p
o = o + p
q = n + o
```

(b) What is Loop unrolling? Explain with example.

(1 + 2 + 3) + 6 = 12

9. Write short note on(any three) **(3 × 4) = 12**
   (i) Token and Lexemes
   (ii) Dynamic storage allocation
   (iii) YACC
   (iv) LEX
   (v) Uniform symbol table
   (vi) Loop jamming.

Department & Section	Submission Link
IT	https://classroom.google.com/c/MjgyNTc1MzEzMzU1/a/MjgyNTc1MzEzNjE0/details