PREVIOUS QUESTIONS – MODEL QUESTIONS – COMPILER DESIGN

1 a. With a neat diagram, explain the different phases of compilation. **(10 Marks)**

b. Explain input buffering strategy, used in lexical analysis phase. **(10 Marks)**

2 a. Write the transition diagram for an unsigned number. **(04 Marks)**

b. Show that the following grammar is ambiguous: $E \rightarrow E + E \mid E * E \mid (E) \mid id$.

Write an unambiguous grammar for the same. **(06 Marks)**

c. Write a recursive descent parser for the grammar : $S \rightarrow cAd$ $A \rightarrow ab/a$ and for the input 'cad' trace the parser. **(10 Marks)**

3 a. Construct the predictive parse table for the following grammar :

$S \rightarrow a \mid \uparrow \mid (T)$

$T \rightarrow T, S \mid S$ **(10 Marks)**

b. Explain the working of a shift reduce parser. **(05 Marks)**

c. Explain handle pruning. Explain the same for the grammar $E \rightarrow E + E \mid E * E \mid (CE) \mid id$ and the input string is id1 + id2 * id3. **(05 Marks)**

4 a. Consider the following grammar :

$S \rightarrow AS \mid b$

$A \rightarrow SA \mid a$

Construct the SLR parse table for the grammar. Show the actions of the parser, for the input string "abab". **(10 Marks)**

b. Construct the CLR parse table for the following grammar :

$S \rightarrow CC$

$C \rightarrow cC \mid d$ **(10 Marks)**

5 a. Define the following with examples :

Synthesized attribute

Inherited attribute

S – attributed definitions

L – attributed definitions. **(08 Marks)**

b. Explain the parser stack implementation of post fix SDT, with an example. **(08 Marks)**

c. For the SDD shown below :

Production	Semantic rules
$L \rightarrow E_n$	L.val = E.val
$E \rightarrow E_1 + T$	E.val = E1.val + T.val
$E \rightarrow T$	E.val = T.val
$T \rightarrow T1 * F$	T.val = T1.val * F.val
$T \rightarrow F$	T.val = F.val
$F \rightarrow (E)$	F.val = E.val
$F \rightarrow$ digit	F.val = digit.lexval

construct the annotated parse tree for 3 * 5 + 4n. **(04 Marks)**

6 a. Explain the following, with an example :
i) Quadraples
ii) Triples
iii) Indirect triples. **(09 Marks)**

b. Write an algorithm for the unification of a pair of nodes in a type graph. **(06 Marks)**

c. Explain syntax directed translation of switch statements. **(05 Marks)**

7 a. What is an activation record? Explain its possible structure. **(08 Marks)**

b. Explain the design goals for garbage collector. **(06 Marks)**

c. Explain the desirable properties of memory manager. **(06 Marks)**

8 a. What is next use information? Write an algorithm to determine the liveness and next use info for each statement in a basic block. Apply the same for the following basic block :

3. T1 = Add(A) – 4
4. T2 = 4 * i
5. T3 = T1 [T2]
6. Sum = Sum + T3
7. I = I + 1
8. If I ≤ 20 go to 3 **(12 Marks)**

b. Generate the intermediate code for the statement: sum = A [i, j] + B [i, j]. Construct DAG and simplify the code. **(08 Marks)**

PREVIOUS QUESTIONS – MODEL QUESTIONS – COMPILER DESIGN

1 a. Explain, with a neat diagram, the phases of a compiler. **(10 Marks)**

b. Construct a transition diagram for recognizing unsigned numbers. Sketch the program segment to implement it, showing the first two states and one final state. **(10 Marks)**

2 a. Explain the left recursion and show how it is eliminated. Describe the algorithm used for eliminating the left recursion. **(06 Marks)**

b. Eliminate left recursion from the grammar:

$S \rightarrow aB \mid aC \mid Sd \mid Se$

$B \rightarrow bBc \mid f$

$C \rightarrow g$ **(02 Marks)**

c. Given the grammar

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

i) Make necessary changes to make it suitable for LL(1) parsing.

ii) Construct FIRST and FOLLOW sets

iii) Construct the predictive parsing table

iv) Show the moves made by the predictive parser on the input (a, (a, a)) **(12 Marks)**

3 a. Obtain a set of canonical LR(0) items for the grammar: **(08 Marks)**

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

b. Is the grammar in Q3(a) SLR(1)? Give reasons. **(04 Marks)**

c. What is handle pruning? Explain with the help of the grammar $S \rightarrow SS+ \mid SS* \mid a$ and input string aaa*a++. Give a bottom-up parse of the given input string. **(08 Marks)**

4 a. Given the grammar :

$S \rightarrow AA$

$A \rightarrow Aa \mid b$

i) Construct sets of LR(1) items

ii) Construct canonical LR(1) parsing table. **(12 Marks)**

b. Write a note on the Parser generator – Yacc. **(04 Marks)**

c. Write the Yacc specification of a simple desk calculator with the following grammar for arithmetic expressions,

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid digit$

Where, the token digit is a single digit between 0 and 9. **(04 Marks)**

5 a. Explain the concept of syntax-directed definition. **(04 Marks)**

b. Consider the context-free grammar given below: **(08 Marks)**

$S \rightarrow EN$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid \text{digit}$

$N \rightarrow ;$

i) Obtain the SDD for the above grammar.

ii) Construct the parse tree, syntax tree and annotated parse tree for the input string 5*6 + 7;

c. Obtain the post-fix SDT for the grammar in Q.5(b) and illustrate the corresponding parser stack implementation. **(08 Marks)**

6 a. Obtain the directed acyclic graph for the expression a + a * (b – c) + (b – c) * d. Also give the sequence of steps for constructing the same. **(06 Marks)**

b. Translate the arithmetic expression a + –(b + c) into quadruples, triples and indirect triples. **(06 Marks)**

c. Explain the syntax-directed translation of switch-statements. **(08 Marks)**

7 a. Describe the general structure of an activation record. Explain the purpose of each item in the activation record. **(06 Marks)**

b. Explain in detail, the strategy for reducing fragmentation in heap memory. **(08 Marks)**

c. Explain briefly the performance metrics to be considered while designing a garbage collector. **(06 Marks)**

8 a. Discuss the issues in the design of a code generator. **(10 Marks)**

b. What are basic blocks and how do you partition a three-address-code into basic blocks? **(05 Marks)**

c. Write the three-address code and construct the basic blocks for the following program segment.

```
Sum = 0;
for (i = 0 ; i < = 10 ; i++)
  Sum = sum + a [i]
```

(05 Marks)

PREVIOUS QUESTIONS – MODEL QUESTIONS – COMPILER DESIGN

1 a. Explain the different phases of a compiler, with a neat diagram. **(08 Marks)**

b. Explain the token generators and token recognizers, with a simple example. **(04 Marks)**

c. Write a Lex program to recognize the string $a^n b$. **(08 Marks)**

2 a. Define the following, with examples :

i) Ambiguous grammar

ii) Derivation tree. **(06 Marks)**

b. Show that the following grammar is ambiguous

$S \rightarrow i\ C\ t\ S \mid i\ C\ t\ S\ c\ S \mid a$

$C \rightarrow b$

Write the unambiguous grammar for the same. **(08 Marks)**

c. Explain with an example, the recursive descent parser with backtracking. **(06 Marks)**

3 a. Bring out the differences between top-down and bottom-up parsing methods. **(03 Marks)**

b. Compute FIRST ( ) and FOLLOW ( ) symbols for the following grammar and find if the grammar is LL(1)

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \epsilon$

$F \rightarrow (E) / id.$ **(08 Marks)**

c. Given the following precedence relation table, parse the string id + id * id

	id	+	*	$
id		.>	.>	.>
+	<.	.>	<.	.>
*	<.	.>	.>	.>
$	<.	<.	<.	

(09 Marks)

4 a. Explain with an example, the stack implementation of a shift reduce parser. **(10 Marks)**

b. Define the following, with examples.

i) Synthesized attribute

ii) Inherited attribute

iii) Annotated parse tree

iv) Dependency graph. **(10 Marks)**

5 a. What is an activation record? Explain the purpose of different fields in an activation record. **(10 Marks)**

b. Explain the following storage allocation strategies.
i) Static allocation
ii) Heap allocation. **(10 Marks)**

6 a. Define 3-address statement and list the types of 3-address statement. **(08 Marks)**

b. Define the terms quadruples, triples and indirect triples. Give their representation for the assignment statement A = B * (C + D) by generating an appropriate 3-address code. **(12 Marks)**

7 a. Explain the issues in the deisgn of a code generator. **(12 Marks)**

b. Generate the code for the following three address statement, using the code generation algorithm

t = a – b
u = a – c
v = t + u
d = v + u
with d live at the end. **(08 Marks)**

8 a. Optimize the following code

```
Product = 0
i = 1
do
 product = product + A[i] *B[i]
        i = i + 1
 while (i < = 20).
```

(12 Marks)

b. Write an algorithm to construct a DAG from a basic block. **(08 Marks)**

PREVIOUS QUESTIONS – MODEL QUESTIONS – COMPILER DESIGN

1 a. Explain with neat diagram, the various phases of a compiler. Mention the input and outpu for each phase. **(08 Marks)**

b. Define static and dynamic scoping. Explain the working and output of the following programming segment if scoping used is static and dynamic:

```
BEGIN
        Boolean b : = true
        Procedure P ;
        BEGIN
                Print (b) ;
        End ;
    BEGIN
        Boolean b : = false ;
        Call P ;
    End ;
END;
```

(04 Marks)

c. With an example, explain the use and coordination between 'LEX' and 'YACC' the compiler writing tools. **(08 Marks)**

2 Consider the grammar:

$E \rightarrow 5 + T \mid 3 - T$

$T \rightarrow V \mid V * V \mid V + V$

$V \rightarrow a \mid b$

a. What is the use of left factoring? Do the left factoring for the above grammar. **(04 Marks)**

b. Write an algorithm to obtain the FIRST and Follow table. Obtain FIRST and Follow table for the above grammar. **(08 Marks)**

c. Write an algorithm to construct the predictive parsing table. Construct predictive parsing table for the above grammar. **(08 Marks)**

3 Consider the grammar:

$S \rightarrow E\#$

$E \rightarrow E - T$

$E \rightarrow T$

$T \rightarrow F \uparrow T$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow i$

a. Write the algorithm to construct basic finite state control m/c for SLR (1) and action α goto functions entries. **(08 Marks)**

b. Construct the following for the above grammar:

i) Basic finite state control.

ii) SLR (1) parsing table containing action and goto function entries. **(12 Marks)**

4 Consider the grammar:

$G \rightarrow S$
$S \rightarrow E = E$
$S \rightarrow f$
$E \rightarrow T$
$E \rightarrow E + T$
$T \rightarrow f$
$T \rightarrow T * f$

when terminal symbols are {=,+,*,f}

a. Write an algorithm to construct finite state control for LR(1) parser. **(08 Marks)**

b. Construct LR(1) finite state control and explain the algorithm to construct parsing table containing action α goto function entries. **(12 Marks)**

PART – B

5 a. With an example, explain the concept of syntax directed definition. **(08 Marks)**

b. Write the grammar and syntax directed definitions for a simple desk calculator and show annotated parse tree for the expression (3+4)*(5+6). **(12 Marks)**

6 a. What is DAG? Construct a DAG for the following expression, $a + a * (b - c) + (b - c) * d$. **(04 Marks)**

b. With an example, explain the various formats of intermediate code. **(10 Marks)**

c. Write quadruple representation for, $a + a * (b - c) + (b - c) * d$. **(06 Marks)**

7 a. Explain the run time storage scheme for C++-language. Give the structure of activation record and explain with suitable example. **(12 Marks)**

b. Explain the design goals for garbage collectors. **(08 Marks)**

8 a. Discuss the following terms:

i) Basic blocks
ii) Next-use information
iii) Flow graph **(10 Marks)**

b. Explain the following code optimization with example:

i) Finding local common sub expression.
ii) Dead code elimination. **(10 Marks)**