1. Which is the most powerful parsing mechanism?

2. A bottom up parser generates  
 a) Right most derivation  
 b) Right most derivation in reverse  
 c) Left most derivation  
 d) Left most derivation in reverse

3. A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammer

S----> xxW ( PRINT "1")

S----> y { print " 2 " }

S----> Sz { print " 3 " )

What is the translation of xxxxyzz using the syntax directed translation scheme described by the above rules ?

4. In which phase normally Type checking is performed?

1. What are the different intermediate code forms?
2. What is a synthesized attribute and inherited attribute?
3. What is the difference between SDD and SDT?
4. What is S-attribute definition and L-attribute definition?
5. What is a handle?
6. What is meant by L, R, 0 in LR(0)?
7. What are the possible conflicts in SLR(1)?

12. IF the base production is in the form of A🡪α.Bβ, a the Look Ahead of the derived production B🡪.γ is \_\_\_\_\_\_\_

1. **Given the following expression grammar:**  
   **E -> E \* F | F+E | F**  
   **F -> F-F | id**  
   **which of the following is true?**   
   (a) \* has higher precedence than +  
   (b) – has higher precedence than \*  
   (c) + and — have same precedence  
   (d) + has higher precedence than \*
2. **Consider the grammar with the following translation rules and E as the start symbol.**  
   **E -> E1 #T {E.value = E1.value \* T.value}**  
   **| T {E.value = T.value}**  
   **T -> T1 & F {T.value = T1.value + F.value}**  
   **|F {T.value= F.value}**  
   **F -> num {F.value = num.value}**  
   **Compute E.value for the root of the parse tree for the expression:2 # 3 & 5 # 6 &4**
3. Generate the three address code for an expression x: = a + b \* c +d ;
4. Which bottom up parser supports even ambiguous grammars also?
5. What is a augmented grammar?
6. What is the parsing table size of LR (0) items?
7. DAG stands for\_\_\_\_\_\_\_
8. Construct DAG for a:=a+(a+a)+(a+a+a+a)
9. What a operator grammar and operator precedence grammar?
10. Consider the grammar G whose CLR parser has n1 states and LALR parser has n2 states. What is the relation between n1 and n2?  
    (A) n1 ≤n2  
    (B) n1 = n2  
    (C) n1 ≥n2  
    (D) None of the above
11. Define annotated parse tree?
12. Generate syntax tree for following expression :

a\*b-c/e+f

1. Regarding the LR(0) and SLR(1) parse tables on which entries we may have the variation
2. Only shift entries c. only reduced entries
3. Both shift and reduced entries d. both reduced and error entries.
4. What is precedence and associatively?
5. YACC stand for\_\_\_\_\_
6. Why Bottom-up parsers called as Shift Reduce parsers?
7. In operator precedence parsing table when will be the 2 operators will get equal precedence i.e. a=b.
8. Which conflict is produced by a state Ik having A🡪α**.** And B🡪γ**.**

**TEN MARK QUESTIONS**

1. Construct SLR(1) Parsing table for the following grammar E->E+T| T; T->T\*F |F; F->(E) | id and show the parsing action for the string id +id +id.
2. Construct a canonical parsing table for the grammar given below

S -> CC

C -> cC | d

1. Consider the grammar

S-> L =R

S->R

L->\*R

L->id

R->L using operator precedence parser, parse the string id= id \* id and also find out the function table.

1. Construct Quadruples, Triples, and Indirect Triples of the following expression:

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

1. Discuss about

i) Postfix Notation 3M

ii) Syntax Trees. 3M

iii) Generate three address code for the following code construct. 6M

1. **a < b or c < d**
2. for(i=0;i<10;i++)

a=a+10