DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER VI [INFORMATION TECHNOLOGY]
SUBJECT: (IT 608) LANGUAGE TRANSLATOR

Examination	**: Second Sessional**	**Seat No.**	**:**
Date	**: 10/02/2014**	**Day**	**: Monday**
Time	**: 12.45 to 2.00**	**Max. Marks**	**: 36**

INSTRUCTIONS:

1. Figures to the right indicate maximum marks for that question.
2. “^” indicates null , “|” is a rule separator , other symbols used carry their usual meanings
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 Do as directed.

(a) In *single pass compiler*, the variable names are inserted into symbol table during ____while in two pass compiler it is during ____. [Scanner/semantic analysis/parser /code generation/ NO phase]. **[1]**

(b) Consider the translation scheme shown below: **[2]**
S → TR , R→ +T {print(‘+’);} R | ^
T → num {print(num.val);}
For an input string ‘8+5+2’, would this translation scheme print “852++”? Justify your answer using annotated parse tree.

(c) Which of the following statements is false? **[1]**
(i)An unambiguous grammar has same leftmost and rightmost derivation
(ii) An LL(1) parser is a top-down parser
(iii) LALR is more powerful than SLR
(iv) An ambiguous grammar can never be LR(k) for any k

(d) In bottom-up evaluation of syntax directed definition, inherited attributes can **[1]**
(i) Always be evaluated.
(ii) Be evaluated only if the definition is L-attributed.
(iii) Be evaluated only if the definition has synthesized attributes.
(iv) Never be evaluated.

(e) Which of the following grammar rules violate the requirements of an operator grammar? P, Q, R are nonterminals, and r, s, t are terminals. Also **justify** your answers. **[2]**
(i)P→QR (ii)P→QsR (iii)P→^ (iv)P→QtRr

(f) Amongst operator precedence parser and LR parser which is better? Why? **[2]**

(g) In a syntax directed definition, ____ attribute is calculated from attributes of children of that node while ______attribute is calculated from attribute of siblings or parent of that node. [S / L/ M / N] **[1]**

(h) __________type of organization in symbol table of block structured languages is similar to *unordered type of* symbol table in non block structured languages. **[1]**
[stack /tree /hash]

(i) Give at least one drawback of tree structured symbol table in non block structured languages. **[1]**

Q.2 Attempt ***Any Two*** from the following questions. **[12]**

(a) Obtain the precedence functions for the following grammar and trace operator precedence parser for the following input: “ id – id % id ”. [6]
E→E – E | E % E | id *[Note: -division “%” has higher precedence then “-” minus.]*

(b) Consider the grammar with the following translation rules and E as the start symbol. [6]
E→ E1#T {E.value=E1.value*T.value}
E →T {E.value=T.value}
T→ T1 & F {T.value = T1.value + F.value}
T → 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.**
Show the annotated tree clearly.

(c) Using examples explain any three symbol table organization techniques for non-block structured languages. [6]

Q.3 (a) Is following grammar suitable for parsing by SLR parser? Justify. **[8]**
E’->E# E→ E sub E sup E | E sub E | E sup E | {E} | c
Note: clearly show the first, follow & closure set elements (if used)

(b) Give SDD to print prefix form of given infix expression. Grammar to describe the valid infix expressions is given below. **[4]**
E → E -T | T T → T*F |F F → id

OR

Q.3 (a) Is following grammar suitable for parsing by SLR parser? Justify. **[8]**
S’ → S# S → qABd A → x | ^ B → y | ^
Parse the string “qxyd”, using the table.
Note: clearly show the first, follow & closure set elements (if used)

(b) Following grammar generates expression formed by applying arithmetic operator * to integer and real constants. When two integers are multiplied the resulting type is integer or real based on operand types. **[4]**
Give SDD to determine type of each expression.
E → E*T|T T → num.num T → num

****