Total No. of Questions :12]

SEAT No. :

P1800

[Total No. of Pages :4

[4859]-203

B.E. (Computer Engineering)

PRINCIPLES OF COMPILER DESIGN

(2008 Course) (Semester - I)

Time : 3 Hours] *[Max. Marks :100*

Instructions to the candidates:

1) ***Answer 3 questions from section I and 3 questions from section II.***
2) ***Answers to the two sections should be written in separate books.***
3) ***Neat diagrams must be drawn wherever necessary.***
4) ***Figures to the right indicate full marks.***
5) ***Assume suitable data, if necessary.***

SECTION - I

Q1) a) Explain role of lexical analyzer in compilation process. Also explain how lexical errors are identified. **[6]**

b) Construct predictive parsing table for following grammar. Verify your answer by showing the parser steps for a valid string of grammar. **[8]**

$P \rightarrow St$

$S \rightarrow q\ ABC$

$A \rightarrow a \mid bb\ D$

$B \rightarrow a \mid \in$

$C \rightarrow b \mid \in$

$D \rightarrow c \mid \in$

c) Discuss how parsing techniques are classified? Support your answer with example. **[4]**

OR

Q2) a) Discuss various design issues related to compiler. **[4]**

b) Construct SLR parsing table for following grammar: **[8]**

$S \rightarrow x\ A\ y \mid x\ B\ y \mid x\ A\ z$

$A \rightarrow a\ S \mid q$

$B \rightarrow q$

Verify your answer by showing parser steps for a valid string of grammar.

c) Explain Recursive Descent Parser with example. Will Recursive Descent Parser work with ambiguous grammar? **[6]**

P.T.O.

Q3) a) What is role of semantic analyzer in compilation process? **[4]**

b) Differentiate between synthesized attributes and inherited attributes. By taking example of declarative statement in 'C' language, explain how inherited attributes are used with bottom-up parser. **[6]**

c) Explain following terms in brief: **[6]**

i) Type checking

ii) Type expression

iii) Type coercion

OR

Q4) a) What is syntax directed translation? What are its advantages? **[4]**

b) Given following grammar for Pascal style declaration. **[8]**

$D \rightarrow L : T$

$L \rightarrow L, id \mid id$

$T \rightarrow integer \mid real$

Add semantic actions for adding type of identifier in the symbol table. Draw the parse tree and annotated parse tree for the sentence

a, b, c : real

c) What is type conversion? How semantic analyzer accomodates type conversion? Explain with example. **[4]**

Q5) a) Explain syntax directed translation scheme for Boolean expression. Generate intermediate code for following statement using the above scheme. p and q or *x* and *y* **[8]**

b) Differentiate between triples and indirect triples. Give proper example. **[4]**

c) Explain the support given by LEX and YACC tools for generating intermediate code. **[4]**

OR

Q6) a) Translate following code fragment into three address statements. Represent these three address statements in quadruple format. **[8]**

b) Write syntax directed scheme for simple assignment statement. Generate intermediate code for statement a = b + c * d using this scheme. **[6]**

c) What is use of marker non terminal in syntax directed translation scheme. **[2]**

SECTION - II

Q7) a) Which are different data structures used for symbol table? Discuss in detail. **[6]**

b) Define activation record. Explain with example the elements of activation record. **[6]**

c) Explain the difference in storage organization and allocation strategies for block-structured and non-block structured languages. **[6]**

OR

Q8) a) Discuss storage organization and allocation strategies. **[6]**

b) Explain run-time management of variable length data. **[6]**

c) Explain following:

Call by Value, Call by Name, Call by Reference **[6]**

Q9) a) With proper example, explain the process of generating code from labeling tree. **[8]**

b) What is DAG? Explain its use in code generation. Generate DAG for following three address statements. **[8]**

$T_1 = A + B$

$T_2 = C + D$

$T_3 = E - T_2$

$T_4 = T_1 - T_3$

OR

Q10)a) What are various issues related to code generation? **[4]**

b) With suitable example explain basic blocks and flow graph. **[4]**

c) Write short notes: Peephole optimization, code generator. **[8]**

Q11)a) Explain fundamental data flow properties. **[8]**

b) Explain following optimizations with examples: **[8]**

i) Common subexpression elimination

ii) Code movement

iii) Variable propagation

iv) Strength reduction

OR

Q12)a) Compare quadruples, triples and indirect triples with respect to their use in code optimization. **[6]**

b) What are induction variables? Explain with example how induction variables help in loop optimization. **[6]**

c) Explain in brief: Live variable analysis. **[4]**

ɛɛɛ