Total No. of Questions :12]

SEAT No. : □

P2015

[Total No. of Pages : 4

[5254]-186

B.E. (I.T.)

COMPILER DESIGN

(2008 Pattern) (Elective - I)

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

Instructions to the candidates:

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

SECTION - I

Q1) a) What do you mean by compiler? Explain analysis-synthesis model of compilation for a = (b + c)*(d – e). **[10]**

b) Explain input buffering with example. What do you mean by sentinels?**[6]**

OR

Q2) a) Write different issues in lexical analysis. Also explain lexical errors and its recovery techniques for each error. **[6]**

b) Explain various compiler construction tools for the compiler design. **[6]**

c) Define tokens, Patterns and lexemes with example for each. **[4]**

Q3) a) Write algorithm for left recursion and for following grammar **[6]**

$S \rightarrow Aa \mid b$

$A \rightarrow Ac \mid Sd \mid \varepsilon$

Eliminate left recursion.

b) For a given grammar **[12]**

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$

Left factor the grammar then find First and Follow and build predictive parsing table, is this LL(1) grammar yes or no.

P.T.O.

OR

Q4) Construct Canonical parsing table for the grammar **[18]**

S' → S

S → CC

C → cC | d

Q5) a) Write translation scheme for checking the types of statements for following grammar **[8]**

S → id: =E

S → if E then S_1

S → while E do S_1

S → S_1; S_2

With this also explain type checking of function with example.

b) Write shorts notes with example for each on **[8]**

i) S-attributed

ii) Inherited attributed

iii) Abstract syntax tree

iv) Dependency graph

OR

Q6) a) Construct syntax tree for a-4+c and directed acyclic graph for a + a * (b - c) + (b - c) * d and differentiate between syntax tree and directed acyclic graph. **[8]**

b) For grammar write down the semantic rules using the Stack of the S/R parser and Left recursive grammar with example. **[8]**

L → E **n**

E → E_1 + T | T

T → T_1* F | F

F → (**E**)

F → digit

SECTION - II

Q7) a) Explain following with suitable example. **[8]**

i) Activation record

ii) Control stack

iii) Binding and storage

iv) Displays.

b) Explain static and Dynamic Scope in detail. Illustrate with example. **[8]**

OR

Q8) a) Explain following parameter passing with proper examples **[8]**

i) Call by value

ii) Call by address

iii) Value result

iv) Copy rule

b) How the records of nested procedures are maintained at run time, explain with the help of neat diagram. (Consider all cases) **[8]**

Q9) a) Explain back-patching with example. **[4]**

b) Write Quadruple, Triple and Indirect Triple representation of following expression d= - (a - b) + (a - c) + (a - c) with explanation. **[12]**

OR

Q10)a) What is Liveness? Explain Liveness calculation with suitable example.**[10]**

b) In register allocation explain following with suitable example **[6]**

i) Control and data flow graph

ii) Interference graph

iii) Spilling

Q11)a) Explain implementation of single and multi-inheritance in compiler design with block diagram. **[10]**

b) How the compiler handles the following types of constructors in object oriented programming? Explain with example. **[8]**

i) Parameterized constructors

ii) Default constructors

iii) Copy constructors

iv) Conversion constructors

OR

Q12)a) Explain implementation of class by compiler with block diagram. **[6]**

b) How overloading and overriding are implemented in compiler. Explain with example. **[8]**

c) Explain Object oriented features in compiler. **[4]**