

**COMPILER CONSTRUCTION**

Time : 3 Hours

Min. Passing Marks : 24

Maximum Marks : 80

**Instruction to Candidates :**

Attempt any five questions, selecting one question from each unit. All questions carry equal marks. (Schematic diagrams must be shown wherever necessary. Any data you feel missing suitably be assumed and stated clearly. Units of quantities used/calculated must be stated clearly.)

**Unit-I**

1. (a) What are the different phases of compiler? Explain them with the help of suitable example. [10]
- (b) Explain the following terms in brief
  - (i) Input buffering.
  - (ii) Functions of lexical analyzer. [3×2=6]

**OR**

1. (a) Consider context free grammar  $S \rightarrow SS+|SS^*|a$ .
  - (i) Show how the string  $aa+a^*$  can be generated by this grammar.
  - (ii) Construct a parse tree for this string.
  - (iii) What language is generated by this grammar? Justify your answer. [8]
- (b) Construct minimum state DFA's for following regular expression.
  - (i)  $(a|b)^*a(a|b)$
  - (ii)  $(a|b)^*a(a|b)(a|b)$
  - (iii)  $(a|b)^*a(a|b)(a|b)(a|b)$  [8]

**Unit-II**

2. Consider the following grammar G:
 
$$E \rightarrow E+T|T$$

$$T \rightarrow TF|F$$

$$F \rightarrow F^*|a|b$$
  - (i) Construct the SLR parsing table for this grammar.
  - (ii) Construct the LALR parsing table. [8×2=16]

**OR**

2. Write down a short note on following:
  - (i) Operator precedence parser for regular expressions.
  - (ii) Difference between bottoms up and top down parsing with suitable example.
  - (iii) YACC error handling in LR parser.
  - (iv) Context free grammar. [4×4=16]

**Unit-III**

3. (a) Give a syntax-directed definition to translate infix

expression into infix Expression without redundant parenthesis. For example, since + and \* Associate to the left,  $((a*(b+c))*(d))$  can be rewritten as  $a*(b+c)*d$ . [10]

- (b) Write simplifications of a simple type checker with example. [6]

**OR**

3. Write a program to translate an infix expression into postfix form. Also writes down syntax directed definition for the same. [16]

**Unit-IV**

4. (a) Explain the various runtime storage management techniques? Explain these with the help of suitable programming example. [10]
- (b) Explain the differences between stack allocation and heap allocation strategies. [6]

**OR**

4. Write short notes on (any two):
  - (i) Symbol table and Dangling References.
  - (ii) Activation Records and Parameter Passing.
  - (iii) Storage allocation strategies. [8×2=16]

**Unit-V**

5. (a) Construct a DAG for the basic block whose code is given below:
 
$$D:=B*C$$

$$E:=A+B$$

$$B:=B*C$$

$$A:=E-D$$
 [10]
- (b) What is peephole optimization? Explain it. [6]

**OR**

5. (a) Explain in brief the various issues of design of a code generator. [8]
- (b) Explain the basic block and control flow graph. [8]