Register No.: ..... Name: .....

SAINTGITS COLLEGE OF ENGINEERING (AUTONOMOUS)

(AFFILIATED TO APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY, THIRUVANANTHAPURAM)

SIXTH SEMESTER B.TECH DEGREE EXAMINATION (S), AUGUST 2023
COMPUTER SCIENCE AND ENGINEERING
(2020 SCHEME)

Course Code : 20CST302

Course Name: Compiler Design

Max. Marks : 100 **Duration: 3 Hours**

PART A

(Answer all questions. Each question carries 3 marks)

1. Differentiate Compiler and Interpreter.
2. What is bootstrapping a compiler?
3. Left factor the following grammar.
   S → x / xy / xyg / xygh/xygha/b
4. Show that the following grammar is ambiguous.
   S→ aSbS |bSaS|ε
5. Explain about handle and handle pruning.
6. How to verify a grammar is operator grammar? Give an example for operator grammar.
7. Distinguish between synthesized and inherited attributes.
8. Describe various fields in an activation record.
9. Write the three-address code sequence for the assignment statement.
   d:=(a-b)+(a-c)+(a-c)
10. What do you mean by machine dependent and machine independent optimization?

PART B

(Answer one full question from each module, each question carries 14 marks)

MODULE I

11. What are the various phases of a compiler? Explain each phase in detail by using the input "a=(b+c)*(b+c)*2". (14)

OR

12. a) Draw transition diagrams that recognizes (7)
    (i) Relation operator
    (ii) Unsigned numbers
    (iii) Identifiers
    b) How is input buffering used in the process of lexical analysis during compiler design? Explain with an example. (7)

MODULE II

13. Construct the predictive parsing table for the following grammar: (14)
S → (L) | a
L → L,S | S
Parse the string "((a,a), (a,a))".

OR

14. a) Construct Recursive Descent Parser for the following grammar. (7)
Prog → { Stmts } Eof {
Stmts → Stmt Stmts id if
Stmts → λ }
Stmt → id = Expr ; id
Stmt → if (Expr) Stmt if
Expr → id Etail id
Etail → + Expr +
Etail → - Expr -
Etail → λ

b) Prove that the following grammar is not LL(1). (7)
S → iEtSS' | a
S' → eS | ε
E → b

MODULE III

15. a) Find the LR(0) items for the following grammar. (7)
E→ E+T|T
T→T*F|F
F→id

b) What are the four actions in shift reduce parsing? (7)
Consider the following grammar.
S→ T L ;
T→ int | float
L→ L , id | id
Parse the input string " int id , id ; " using a shift-reduce parser.

OR

16. Construct CLR(1) parsing table for the following grammar and parse the string "*i=*i" . (14)

S→L=R|R
L→*R|i
R→L

MODULE IV

17. a) Discuss the various storage allocation strategies in detail. (6)

b) Translate the following expression to quadruple, triple and indirect triple. (8)

$$a = b \times -c + b \times -c$$

OR

18. a) Write an SDD for a simple desk calculator and draw the annotation parse tree for the evaluation of the expression (2*5)+(3*4)n. (11)

b) Write SDT for converting infix to postfix expression. (3)

MODULE V

19. a) Describe the various techniques used by compilers to optimize code while preserving the basic structure of the code, specifically focusing on the structure-preserving transformations for basic blocks, and how they impact the control flow and data dependencies within a basic block? (7)

b) Elucidate three loop optimization techniques. Apply Loop optimization to the following code. (7)

```
i= 0
a:=n-3;
if i< a then loop else end
label loop
b:= i -4
c:= p + b
d := m[c]
e := d-2
f:= i – 4
g:= p + f
m[g]:= e
i = i +1
a:= n- 3
if i < a then loop else end
label end
```

OR

20. a) What are the key issues that need to be considered in the design of a code generator, and how do these issues impact the quality, performance, and compatibility of the generated code? (9)

b) Develop a DAG and optimal target code for the expression. (5)
x = ((a + b) / (b-c)) – (a + b) * (b-c) +f.

****