Name: **Enrolment No:**	

UPES

End Semester Examination, December 2023

Course: Compiler Design **Semester: V**
Program: B.Tech (CSE) All Specialization **Time : 03 hrs.**
Course Code: CSEG 3015 **Max. Marks: 100**

Instructions: Attempt all questions, however internal choice is mentioned.

SECTION A (5Qx4M=20Marks)

S. No.		**Marks**	**CO**

Q 1	Consider the following C program and find out the number of tokens using lexical analyzer of compiler.
	main () {
	int *a, b;
	b = 10;
	a = &b;
	printf("%d%d", b, *a);
	b= */*pointer*/b; }
		4	**CO1**

Q2	Consider the regular language $L = (111+11111)^*$, where $\Sigma \in \{1\}$, Construct the DFA and find out the minimum number state for this language.	**4**	**CO2**
Q3	Consider the translation scheme shown below: S→T R R→+T {print ('+');} R | ϵ T→ num {print (num.val);} Here num is a token that represents an integer and num.val represents the corresponding integer value. For an input string **'9+5+2'**, what this translation scheme prints?	**4**	**CO4**
Q4	Comprehend the comparison between synthesized attributes and inherited attributes with example.	**4**	**CO4**
Q5	State the definition of Context Free Grammar. Also, list the differences between all the types of grammars with example.	**4**	**CO1**

SECTION B (4Qx10M= 40 Marks)

S. No.		**Marks**	**CO**

Q6	Generate three address code for the following code-
	while (A < C and B > D) do
	    if A = 1 then C = C + 1
	    else
	      while A <= D
	        do A = A + B
		10	**CO4**

Q7	Consider the following grammar:- E → E + T / T T → T * F / F F → (E) / id Construct the first and follow sets for the grammar. Also design a LL(1) parsing table for the grammar.	**10**	**CO2**
Q8	Code optimization is an important phase of compiler. Explain all the type of code optimization with proper example.	**10**	**CO5**
Q9	Write the rule to non-determinism in a grammar. Do left factoring (if present) in the following grammar: S → bSSaaS / bSSaSb / bSb / a **-- OR—** Consider the following grammar- E → EAE | id A → + | x Construct the operator precedence parser and parse the string id + id x id.	**10**	**CO3**
SECTION-C (2Qx20M=40 Marks)			
Q10	Consider the following grammar: S→ Aa / bAc / Bc / bBa A→ d B→ d Conduct the CLR and LALR parsing for the given grammar to obtain the parsing tables, if possible. Otherwise, specify the problem with justification.	**20**	**CO4**
Q11	Consider the following basic block- B10: S1 = 4 x I S2 = addr(A) – 4 S3 = S2[S1] S4 = 4 x I S5 = addr(B) – 4 S6 = S5[S4] S7 = S3 x S6 S8 = PROD + S7 PROD = S8 S9 = I + 1 I = S9 If I <= 20 goto L10 • Draw a directed acyclic graph and identify local common sub-expressions. • After eliminating the common sub-expressions, re-write the basic block. **----OR---** Define the following terms with example: i. Lex Program with example ii. Activation record iii. Parse Tree vs Syntax Tree with example iv. Peep-hole Optimization	**20**	**CO5/ CO3**