U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

January / February 2025 Semester End Main Examinations

Programme: B.E. **Semester: V**
Branch: Computer Science and Engineering **Duration: 3 hrs.**
Course Code: 22CS5PCCPD **Max Marks: 100**
Course: Compiler Design

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - I	CO	PO	Marks
1	a)	Illustrate the structure of the compiler with a neat diagram and apply the same for the expression position=initial +rate*60	CO1	PO1	**10**
	b)	Write a Lex program to count the number of words in a given input sequence	CO1	PO1	**6**
	b)	Write the transition diagram for recognizing signed numbers	CO1	PO1	**4**
		OR			
2	a)	Outline the differences between Interpreter and Compiler	CO1	PO1	**4**
	b)	Write a program to check if the input sentence ends with any of the following punctuation marks (?, fullstop, !) using appropriate meta character	CO1	PO1	**6**
	c)	Explain the concept of input buffering in lexical analyzer with its implementation code.	CO1	PO1	**10**
		UNIT - II			
3	a)	Write a YACC program to implement a basic calculator	CO2	PO2	**5**
	b)	Apply predictive parsing for following grammar **S→AB** **A→xA / B** **B→yxB| z** Also check if the string xyxz is accepted or rejected.	CO2	PO2	**10**
	c)	Develop a recursive descent parser for the grammar : **S→cad** **A→ab|a** and for the input **"cad"** trace the parser.	CO2	PO2	**5**
		OR			
4	a)	Eliminate Left recursion from the following grammar. E -> E + T | E-T | T T -> T * F | T/F | F	CO1	PO1	**5**

			F-> (E) | id			
		b)	Consider the following grammar **S →A+B / (S)** **A→ aAa | b** **B→ aBa | a** Build the SLR parse table for the above grammar with transition diagram.	*CO2*	*PO2*	**10**
		c)	Eliminate left factoring for the following **S->bSSaaS|bSSaSb|bSb|a** **P->a|ab|abc|abcd**	*CO2*	*PO2*	**5**
			UNIT - III			
	5	a)	Construct schematic rules for the simple type declaration grammar D→ T L T→ int / float L→ L,id / id Write a dependency Graph and annotated parse tree for the input string **float id1,id2**	*CO2*	*PO2*	**10**
		b)	Write semantic rules for a simple desk calculator. and construct an annotated parse tree for the following expression given using SDD for a simple desk calculator. **(3+4)*(5+6)n**	*CO2*	*PO2*	**10**
			OR			
	6	a)	Write syntax Directed Definition for a desk calculator having operations +, * for a top-down parser. Show the dependency graph for the input 5+4*3.	*CO2*	*PO2*	**10**
		b)	Give SDT for simple arithmetic expression using top down Approach (L-attributed definition) for a-4+c.	*CO2*	*PO2*	**10**
			UNIT - IV			
	7	a)	Explain the various three-address statements and apply this to find quadruple and triples and indirect triples for a given expression (a-b)*(c+d)-(x+y)	*CO3*	*PO3*	**10**
		b)	Write DAG and the three address code for the following i) x=Z*y + sin(Z*x) Z=x/Z ii)b=(((a + a) + (a + a)) + ((a + a) + (a + a)))	*CO3*	*PO3*	**10**
			OR			
	8	a)	Write semantic rules for flow of control statements and Boolean expressions	*CO3*	*PO3*	**12**

		b)	Analyze the following code snippet and give its equivalent three address code
			while(a<c and b>d)
			{
			if a=1
				3	*2*	**8**

			then c=c+1
			else
			while (a <= d)
			a=a+b
			}			

			UNIT - V			

	9	a)	Construct the Control Flow Graph for the given code:
			i = 0; j = 0; k = l;
			while (i < n1 && j < n2)
			{
			if (L[i] <= R[j])
			{
			arr[k] = L[i]; i++;
			}
			else
			{
			arr[k] = R[j]; j++;
			}
			k++;
			}
				CO3	*PO3*	**12**

		b)	Explain the design issues in code generator	*CO3*	*PO3*	**8**
			OR			
	10	a)	Generate machine code for the following three address code i) x=y-z ii) b=a[i] iii)*p=x	*CO3*	*PO3*	**6**
		b)	Outline any 3 addressing modes with example. Calculate program instruction cost for the following code i) LD R0,R1 ii) LD R1 ,*100(R2)	*CO3*	*PO3*	**10**
		c)	Outline the steps involved in partitioning three address code into basic blocks	*CO3*	*PO3*	**4**

****