QP Code: RA22BTECH424 | Reg. No | | | | | | | | | | | AR21/22

Gandhi Institute of Engineering and Technology University, Odisha, Gunupur
(GIET University)

B. Tech (Sixth Semester – Regular/Supplementary) Examinations, April 2025

21BCSPC36004/22BCSPC36004/21BCMPC36004/22BCMPC36004 /21BCDPC36004/22BCDPC36004 – Compiler Design

(CSE/CSEDS/CSEAIML)

Time: 3 hrs | Maximum: 70 Marks

Answer ALL questions

(The figures in the right hand margin indicate marks)

PART – A **(2 x 5 = 10 Marks)**

		CO #	Blooms Level
Q.1.	Answer ***ALL*** questions		
a.	Define the terms *lexeme*, *token*, and *pattern* in the context of lexical analysis.	CO1	K1
b.	What is an ambiguous grammar in the context of compiler design? Explain how ambiguity in a grammar can be identified and eliminated.	CO1	K2
c.	What is a shift-reduce conflict in parsing? Describe when and why it typically occurs during syntax analysis.	CO2	K2
d.	What is an L-attributed definition in compiler design? Describe its characteristics and significance in syntax-directed translation.	CO3	K2
e.	Describe Back patching.	CO4	K2

PART – B **(15 x 4 = 60 Marks)**

		Marks	CO #	Blooms Level
	Answer ***all*** the questions			
2. a.	Explain the different phases of a compiler in detail. Describe the input and output of each phase and how the data flows from one phase to another. Use a diagram to support your explanation	8	CO1	K1
b.	Draw the NFA with ε move for the RE= aa*|bb* and convert to DFA.	7	CO1	K2
	(OR)			
c.	Discuss the concept of input buffering in lexical analysis. Why is it essential for efficient scanning? Explain the two-buffer scheme with a neat diagram	8	CO1	K1
d.	Write the Algorithm to construct predictive parsing table.	7	CO1	K2
3.a.	Check the given grammar is LR(0) or not P={ S→AA, A→AA|aA|b }	10	CO2	K3
b.	Given the following grammar: P={ E→E+T|T, T→T*F|F, F→ 1|2|3 } Write the SDD to Evaluate the expression 2+3*4 and represent the annotated tree.	5	CO2	K2
	(OR)			
c.	Check the given grammar is CLR or not P={ S→AaAb|BaBa, A→ ε, B→ ε }	10	CO2	K3
d.	Consider the following grammar- E → E + E | E * E | id Construct Operator Precedence Function table.	5	CO2	K2
4.a.	Define TAC. write the Triple for the instruction a=X[i][j]	8	CO3	K2
b.	Explain the different components or sections of an Activation Record in Compiler Design.	7	CO3	K1
	(OR)			

c.	Define SDD. Write the SDD for Boolean expression B→ B1||B2	8	CO3	K2
d.	Define DAG. Represent the DAG for instruction a + a * (b-c) + (b-c) * d	7	CO3	K2
5.a.	Why is code optimization important in compiler design? Explain peephole optimization with suitable examples to illustrate its application.	8	CO4	K1

b.	Define a basic block in the context of compiler design. Identify the basic blocks in the following code and draw the corresponding control flow diagram,
	x, y = 0;
	i = 0;
	while (i <= 5) {
	x = x + 2;
	i++;
	}
	y = y + 1;
		7	CO4	K2

(OR)

c.	Describe, step by step process of code generation for the expression: x=(a+b)-c	8	CO4	K2
d.	What is control flow optimization in compiler design, and how does it improve the efficiency of the generated code? Provide examples of common techniques used in control flow optimization.	7	CO4	K2

--- End of Paper ---