Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science & Engineering
Final Examination, Fall 2025
Course Code: CSE313, Course Title: Compiler Design
Level: 3 Term: 1 Batch: 65

Time: 02:00 Hrs. **Marks: 40**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1.	***a)***	Consider the following grammar to produce LR(0) parser and Canonical Table. **A → B | AB** **B → i | k | (A)** **C → j | kD** **D → m**	**[8]**	**CO2**
	b)	Considering the following code segment: Identify and explain which code optimization technique can be applied to optimize this code. Clearly state the reason for applying the optimization. c = a * b x = a ... d = x * b + 4	**[2]**	
2.	***a)***	Develop FIRST and FOLLOW functions for the following production rules and determine the LL(1) Parsing Table: **X → XSb | Sa | b** **S → Sb | Xa | a**	**[8]**	**CO2**
	b)	Apply the **Left Factoring** technique to the grammar below to remove ambiguity and make it suitable for a predictive parser. **D → declaration | declaration list; declaration**	**[2]**	
3.	***a)***	Consider the following expression and answer the questions. **if a < b and x > z then x = y + z** **else goto L12** **L12: -12** i. Represent the expression using Three Address Code (TAC). ii. Represent the expression using Quadruples. iii. Implement a Triples data structure to represent the expression.	**[6]**	**CO3**
	b)	Consider the following arithmetic expression: **a + a + a + a – a + a + a + b** i. Construct the Syntax Tree for the expression. ii. Construct a Directed Acyclic Graph (DAG) for the expression.	**[4]**	

4. Consider the following three address codes and answer the following questions: **CO3**

1: t1 = a + b	21: i = 0	41: C[j] = t17
2: t2 = c - d	22: s = 0	42: L14: j = j + 1
3: if t1 > t2 goto L1	23: L7: if i >= n goto L8	43: goto L11
4: t3 = t1 * 2	24: t13 = A[i]	44: L12: k = 0
5: t4 = t2 + 5	25: s = s + t13	45: L15: if k >= p goto L16
6: goto L2	26: if t13 < 0 goto L9	46: t18 = D[k]
7: L1: t5 = t1 - t2	27: t14 = t13 * 2	47: if t18 > 100 goto L17
8: t6 = t5 * 3	28: goto L10	48: t19 = t18 * 4
9: if t6 < 0 goto L3	29: L9: t14 = -t13	49: goto L18
10: t7 = t6 + 10	30: L10: B[i] = t14	50: L17: t19 = t18 / 2
11: goto L4	31: i = i + 1	51: L18: E[k] = t19
12: L3: t7 = -t6	32: goto L7	52: if t19 < 0 goto L19
13: t8 = t7 + 1	33: L8: j = 0	53: t20 = t19 + q
14: L4: t9 = t7 + t2	34: L11: if j >= m goto L12	54: goto L20
15: t10 = t9 - t4	35: t15 = C[j]	55: L19: t20 = q - t19
16: if t10 == 0 goto L5	36: if t15 == 0 goto L13	56: L20: F[k] = t20
17: t11 = t10 + 1	37: t16 = t15 + 1	57: k = k + 1
18: goto L6	38: C[j] = t16	58: goto L15
19: L5: t11 = t10 - 1	39: goto L14	59: L16: result = s + t12
20: L6: t12 = t11 * t11	40: L13: t17 = t15 - 1	60: return result

Answer the following questions from the above three address code-

a) i. Which lines in the code qualify as leaders by leader selection rule 2?
ii. Is / Are there any instruction(s) designated as leaders more than twice?
iii. What is the total number of basic blocks identified in the code? **[6]**

b) Draw the **flow graph** for the above mentioned instructions. **[4]**