ourdiu.com | Shared by Chand

Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science & Engineering
Final Examination, Summer 2025
Course Code: CSE313, Course Title: Compiler Design
Level: 3 Term: 1 Batch: 64

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. **S → N@D.T** **N → a |b** **D → Gm |m** **G → e | y | g** **T → c | d**	**[8]**	**CO2**

	b)	`int main() {`
		`  int x = 5, y = 10;`
		`  int p = x + y;`
		`  int q = x + y;`
		`  int z = p * 2;`
		`  int r = z;`
		`  int w = r + 1;`
		`  r = 20;`
		`  return w;`
		`}`
		Explain the techniques of **Code Optimization(s)** that can be applied to the code and rewrite the code after applying these optimization(s) to improve its efficiency.
			[2]	

2.	***a)***	Develop FIRST and FOLLOW functions for the following production rules and determine the LL(1) Parsing Table: **S → S+M | M** **M → M*R | R** **R → aSb | int**	**[8]**	**CO2**
	b)	State the rules of the FIRST() and FOLLOW() functions.	**[2]**	
3.	***a)***	Consider the following arithmetic expression: **(a + b) * (c - d) + e * f - (g + h) * i** Apply intermediate code generation techniques to evaluate the expression. i. Apply your knowledge to write the Three Address Code (TAC) for the expression. ii. Generate the corresponding Quadruples representation. iii. Analyze the structure and produce the Indirect Triples representation where you have 27, 29, 31, 33, 67, and 91 available memory in STACK.	**[6]**	**CO3**

ourdiu.com

ourdiu.com | Shared by Chand

	b)	Consider the following arithmetic expression: **a + b * (c - d) - e + b * (c - d)** i. Construct the Syntax Tree ii. Construct the DAG (Directed Acyclic Graph)	**[4]**	
4.				**CO3**

1	i = m-1	16	$t_7 = 4 * i$
2	j = n	17	$t_8 = 4 * j$
3	$t_1 = 4 * n$	18	$t_9 = a[t_8]$
4	$v = a[t_1]$	19	$a[t_7] = t_9$
5	i = i+1	20	$t_{10} = 4 * j$
6	$t_2 = 4 * i$	21	$a[t_{10}] = x$
7	$t_3 = a[t_2]$	22	goto (5)
8	if $t_3 < v$ goto (5)	23	$t_{11} = 4 * i$
9	j = j-1	24	$x = a[t_{11}]$
10	$t_4 = 4 * j$	25	$t_{12} = 4 * i$
11	$t_5 = a[t_4]$	26	$t_{13} = 4 * n$
12	if $t_5 > v$ goto (9)	27	$t_{14} = a[t_{13}]$
13	if i>=j goto (23)	28	$a[t_{12}] = t_{14}$
14	$t_6 = 4 * i$	29	$t_{15} = 4 * n$
15	$x = a[t_6]$	30	$a[t_{15}] = x$

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]**	