Enrolment No./Seat No.:

GUJARAT TECHNOLOGICAL UNIVERSITY

M.SC INTEGRATED - SEMESTER - VI EXAMINATION - SUMMER 2025

Subject Code: 1360304 **Date: 16-05-2025**

Subject Name: Compiler Design

Time:10:30 AM TO 01:00 PM **Total Marks: 70**

Instructions

1. Attempt all questions.

2. Make suitable assumptions wherever necessary.

3. Use of simple and non-programmable scientific calculators are permitted.

			Marks
Q.1	**(a)**	What is compiler? What is front-end and back-end of compiler?	**03**
	(b)	Give difference between compiler and interpreter.	**04**
	(c)	Explain input, output and action performed by each phases of compiler with example.	**07**
Q.2	**(a)**	Define the terms: token, pattern, lexems.	**03**
	(b)	Give difference between syntax tree and parse tree with example.	**04**
	(c)	Explain input buffering methods in detail.	**07**
		OR	
	(c)	Draw the DFA for the regular expression (a|b)*abb using set construction method only.	**07**
Q.3	**(a)**	What is the role of parser in compiler?	**03**
	(b)	Differentiate Top Down Parsing and Bottom up parsing.	**04**
	(c)	Show the following grammar is LR(1) but not LALR(1). S->Aa | bAc | Bc | bBa A->d B->d	**07**
		OR	
	(a)	Describe Ambiguous Grammar with example.	**03**
	(b)	Explain error recovery strategies used by parser.	**04**

(c) Construct SLR parsing table for the following grammar : **07**

S→(L)|a
L→ L,S | S

Q.4 **(a)** Write a short note on Symbol table management. **03**

(b) What is Left Recursion? Give an example for eliminating the same. **04**

(c) Write a short note on Error Detection and Recovery. **07**

OR

(a) Draw a DAG for expression: a + a * (b – c) + (b – c) * d. **03**

(b) What is inherited attribute? Explain with suitable example. **04**

(c) Classify the errors and discuss the errors in each phase of Compiler. **07**

Q.5 **(a)** Explain Basic Block and Flow Graph with example. **03**

(b) Explain Quadruples, Triple, Indirect Triple with an example. **04**

(c) Explain heap management with the help of an example. **07**

OR

(a) List the generic issues in the design of code generators. **03**

(b) Explain peephole code optimization technique. **04**

***

(c) Explain stack allocation of space with the help of an example. **07**