VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

REG.NO.:

SLOT: A1+TA1

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
FALL SEMESTER 2025-2026

Programme Name & Branch : **B.Tech CSE with All Specialization**
Course Code and Course Name : **BCSE307L - Compiler Design**
Faculty Name(s) : Prof. MUTHUNAGAI S U, Prof. KRISHNARAJ N, Prof. NIVITHA K, Prof. ISLABUDEEN M, Prof. VISWANATHAN A, Prof. SUGANTHINI C, Prof. KANAGARAJ R, Prof. KATARI BALAKRISHNA, Prof. LAKSHMI S, Prof. BHAWANA TYAGI, Prof. BASKARAN P , Prof. SENDHIL KUMAR K.S, Prof. NAGA PRIYADARSINI R, Prof. VISHNU PRIYA A, Prof. BHUVANESWARI M

ClassNumber(s): VL2025260101612,1590,1614,1619,1579,1627,1592,1581,1610,1636, 1597,1608,1632,1584,1587
Date of Examination : **17.08.2025 FN**
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
  Course Outcomes:
  CO1: Apply the skills on devising, selecting, and using tools and techniques towards compiler design
  CO2. Develop language specifications using context free grammars (CFG).

Q. No	Question	M	CO	BL

1.	Explain the phases of compiler in detail and analyse how the following code snippet is processed in each phase:
	if(age>18)
	{
	printf("Eligible to vote");
	}
	else
	{
	printf("Not eligible to vote");
	}
		10	1	3

2.	Construct DFA for the regular Expression $(a^+\|b^+c^*)^*\varepsilon$ (abc) using direct method with all the necessary steps	10	1	3
3.	a) Construct top-down parser using recursive approach for the following grammar with all the necessary steps $A \rightarrow Ax \mid y$ $B \rightarrow cde \mid cdf$	7	2	3

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
FALL SEMESTER 2025-2026

	C → Cab| Cac | d The starting Symbol for the grammar is 'A'			
	b) Check whether the given string is parsed by using above grammar (refer 3a)) or not "yxxxcdfdabacab" with the steps	3		
4.	Consider the Grammar G as follows: S → A A → DBC / BC C → c / ε D → a / d B → Bb / ε Apply LL(1) parsing and construct the parsing table for the above grammar and justify it whether it is LL(1) or not using string "abbc" and the starting State for the above grammar is "S"	10	2	3
5.	Construct an Operator Precedence Parser for the following grammar. Also, build the operator precedence relation table and the operator precedence function table. Show the parsing steps for the input string: "i - i / i" Expr → Expr-Expr Expr → Expr / Expr Expr → (Expr) Expr → i	10	2	3

****